Generating Automated Meeting Summaries

Dissertation
zur Erlangung des Grades
Doktor der Ingenieurswissenschaften (Dr.-Ing.)
der Naturwissenschaftlich-Technischen Fakultät I
der Universität des Saarlandes

vorgelegt von

Thomas Kleinbauer

Saarbrücken, 30.11.2011

Tag des Kolloquiums: 23.12.2011
Dekan der Fakultät: Prof. Holger Hermanns
Mitglieder des Prüfungsausschusses: Prof. Antonio Krüger, Prof. Wolfgang Wahlster, Prof. Manfred Pinkal, Dr. Jan Alexandersson

Eidesstattliche Versicherung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit selbstständig und ohne Benutzung anderer als der angegebenen Hilfsmittel angefertigt habe. Die aus anderen Quellen oder indirekt übernommenen Daten und Konzepte sind unter Angabe der Quelle gekennzeichnet. Die Arbeit wurde bisher weder im In- noch im Ausland in gleicher oder ähnlicher Form in einem Verfahren zur Erlangung eines akademischen Grades vorgelegt.

Melbourne, den 29. November 2011

Danksagung

Zunächst möchte ich mich recht herzlich bei meinem Doktorvater, Prof. Wahlster, für die Möglichkeit bedanken, unter seiner Betreuung an einem interessanten und neuen Forschungsthema zu arbeiten. Die ausgezeichneten Besprechungen haben entscheidend dazu beigetragen, diese Arbeit in ihre heutige Form zu bringen.

Ebenfalls bedanken möchte ich mich bei den Professoren Pinkal und Krüger für ihre Bereitwilligkeit, als Berichterstattender bzw. Vorsitzender des Prüfungsausschusses zu fungieren.

Ganz besonderen Dank gilt meinen beiden ehemaligen Kollegen Tilman und Jan. Vielen, vielen Dank für die Zeit und Mühe, die Ihr investiert habt, um mir durch Diskussionen und Anregungen zu helfen, sowie für die aufmunternden Worte, wenn diese nötig waren.

Vielen Dank auch an Dominik Heckmann als mein verlängerter Arm in Deutschland. Ohne ihn könnte das hier niemand lesen.

Den Mitarbeitern des FRAMENET-Teams sei auch ein herzliches Wort des Dankes ausgesprochen, ganz besonders Chuck Fillmore und Collin Baker. Josef Ruppenhofer war stets ein geduldiger Ansprechpartner auf deutscher Seite, wenn es um das Thema Framesemantiken ging. Vielen Dank! Dasselbe gilt für Miriam R.L. Petruck in Berkeley.

Gabriel Murray bin ich für zahlreiche Diskussionen zum Thema „automatisches Erzeugen von Zusammenfassungen" zu großem Dank verpflichtet.

Meinen früheren Kollegen am DFKI sei ebenfalls gedankt für viele fruchtbare Gespräche und eine hervorragende Forschungsumgebung. Ganz besonders möchte ich mich bei Germi bedanken, dessen Hilfsbereitschaft ich sehr zu schätzen weiss.

Ingrid Zukerman und Arun Mani bin ich für die Hilfe bei der Evaluation zu großem Dank verpflichtet.

All meinen Freunden, die die letzten Jahre nicht viel von mir gehabt haben, sei ebenfalls ein Wort des Dankes gesagt für ihre Geduld und Unterstützung, besonders Jörg, Michaela, Yasemin und Alex.

Zu guter Letzt wäre diese Arbeit niemals möglich gewesen ohne die Liebe und Unterstützung meiner Eltern und meiner Familie. Liebe Mama, lieber Papa, vielen lieben Dank – für alles.

Acknowledgments

First and foremost, I would like to express my gratitude to my thesis supervisor, Prof. Wahlster, for the possibility to research an interesting and novel topic under his supervision. The outstanding meetings have helped tremendously to form this thesis into its final shape.

Likewise, I would like to thank Prof. Pinkal and Prof. Krüger for accepting their respective roles in the graduation committee.

Special thanks go to my former co-workers Tilman and Jan. Thank you so much for the time and effort you invested to help me through tips and discussions, and also for the motivating words whenever they were necessary.

Many thanks to Dominik Heckmann who supported me with remote tasks in Germany. Without him, no-one could read this.

I would also like to express a special word of gratitude to the members of the FrameNet team, especially Chuck Fillmore and Collin Baker. Josef Ruppenhofer has always been a patient helper in Germany whenever questions related to Frame Semantics arose. Thanks a bunch! The same goes to Miriam R.L. Petruck in Berkeley (look: I'm PhinisheD!)

Gabriel Murray deserves great gratitude for many conversations on meeting summarization.

I would also like to thank my former colleagues at DFKI for many fruitful discussions and an outstanding research environment. In particular, I would like to thank Germi, whose willingness to help I appreciate very much.

I am very grateful to Ingrid Zukerman and Arun Mani for their help with the evaluation.

A word of gratitude goes to all my friends who have not seen a lot of me in recent years, for their patience and support, especially to Jörg, Michaela, Yasemin und Alex.

And last but not least, this thesis would never have been possible if it wasn't for the love and support of my parents and my family. Dear mom, dear dad, thank you so much–for everything.

Zusammenfassung

Die vorliegende Arbeit stellt einen neuartigen Ansatz zur Generierung abstraktiver Zusammenfassungen von Gruppenbesprechungen vor. Während automatische Textzusammenfassungen bereits seit einigen Jahrzehnten erforscht werden, liegt die Neuheit dieser Arbeit vor allem in der Anwendungsdomäne (Gruppenbesprechungen statt Textdokumenten), sowie der Verwendung eines lexikalisierten Repräsentationsformulism auf der Basis von Frame-Semantiken, der es erlaubt, Zusammenfassungen abstraktiv (statt extraktiv) zu generieren. Wir argumentieren, dass abstraktive Ansätze für die Zusammenfassung spontansprachlicher Interaktionen besser geeignet sind als extraktive.

Die Arbeit beginnt mit einer Motivation des Forschungsgegenstands und der Beschreibung der zentralen Forschungsfragen. Anschliessend wird der Begriff der „Zusammenfassung" generell diskutiert und verschiedene Dimensionen von Zusammenfassungen verglichen. Es folgt eine Übersicht über verwandte Arbeiten zum Thema „Generierung von Zusammenfassungen". Die für den vorgestellten Ansatz notwendigen Theorien und Datensätze werden anschließend eingeführt. Danach wird die Architektur des für diese Arbeit implementierten MEESU-Systems erläutert, und die Theorie und Umsetzung der einzelnen Komponenten vorgestellt. Das System wurde mittels eines neu entwickelten Verfahrens evaluiert, welches im Anschluss diskutiert wird. Die Arbeit schließt mit einer Zusammenfassung und einer Diskussion möglicher Ansatzpunkte für zukünftige Forschungsarbeiten.

Abstract

The thesis at hand introduces a novel approach for the generation of abstractive summaries of meetings. While the automatic generation of document summaries has been studied for some decades now, the novelty of this thesis is mainly the application to the meeting domain (instead of text documents) as well as the use of a lexicalized representation formalism on the basis of Frame Semantics. This allows us to generate summaries abstractively (instead of extractively).

The thesis begins with an overall motivation of the research domain, and a description of the central research questions. After that, the notion of a "summary" is discussed in general, and different dimensions of summaries are compared, before we give a broad overview over related work in the field of automatic summarization. Then, we introduce the necessary theories for this approach and the data sets used. Following that, we discuss the architecture of the MEESU system which has been developed in the course of this work, as well as the theory and implementation of the contained components. The system has been evaluated using a novel extrinsic evaluation approach which is detailed next. The thesis concludes with a summary and a discussion of possible points for future work.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A good part of our everyday life we spent together with other people, from casual gatherings among friends to quality time in the family to business meetings. While private activities may primarily serve recreation and fun, meetings in the work life are as crucial an instrument for coordination in collaborative work environments as they are for business-to-business communication. Modern technology, such as telephone- or video conferences, email threads, instant messaging etc. have become valuable additions in these field, but have yet to reach the naturalness and efficiency of face-to-face meetings.

One reason that remote meetings by phone or video connection are used as an alternative to face-to-face meetings is the effort in cost and time required to bring people from different places to one location. Consequently, one can argue that meetings "in person" thus become an even more valuable resource and it is especially important that the outcome of such meetings are of high quality. This notion is two-fold. It refers to concrete results achieved during a meeting on the one hand, such as, decisions, action items assigned, problems identified, etc., but, on the other hand, also to methods that ensure that these results are recorded appropriately for later access. A number of different strategies exist to support the latter goal.

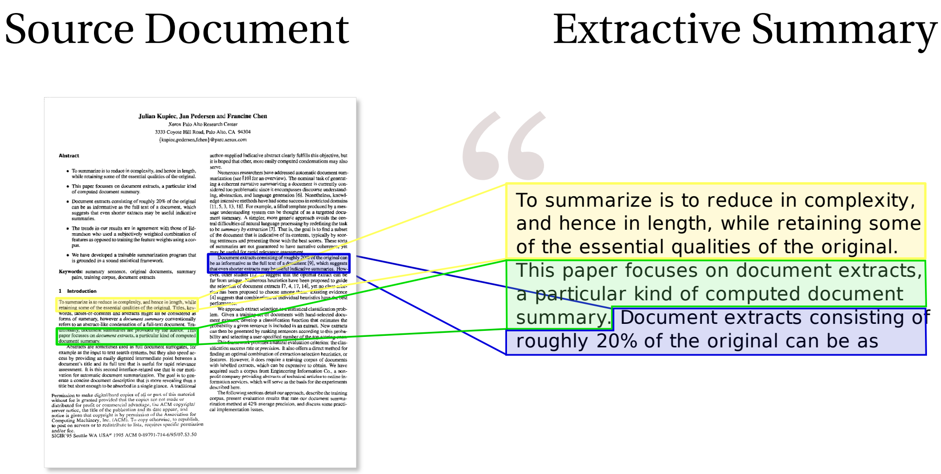
Where the technological and financial means are available, meetings can be recorded audio-visually in full length and stored in a multimedia database. This way, the exact contents can be retrieved at any point in time after the meeting was recorded. However, such an approach brings about an enormous technical and financial burden and is and not yet general practice in the business world. Furthermore, even if recording and storing every meeting will become feasible in the future, the question arises how to access a steadily growing archive of recorded meetings efficiently. Given a specific information need this may imply finding the relevant meetings in the database as a first step. A second step would then be to consult the audio-visual recordings to extract the relevant parts that satisfy the user interest. Depending on the length of the returned meetings, this can become a very timely procedure and, where the information need is complex and further technological support lacking, impractical altogether.

Summaries are a proven tool to quickly assess the gist of a document. We find different forms of summaries at various occasions in our everyday lives. For instance, newspaper articles often feature a small abstract above the full article. But summaries are not restricted to documents, summaries of other media (e.g., movie plots in a television guide) or live events (e.g., a review of a concert, a report of a sports match, etc.) are equally ubiquitous. For meetings, summaries typically come in the form of hand-written meeting minutes.

Given the benefits of summaries, it seems they are not utilized as frequently as one would expect. For instance, while important meetings such as business within a company may be documented with official minutes, less formal meetings often times are not. Would it not be more useful if simply *every* meeting were recorded with at least a short summary document that could later on be used as a collective memory and as the basis for information retrieval? A likely answer to that question is the inherent time and effort required to create high quality summaries. In meetings, the task of writing minutes is sometimes delegated to a secretary, if available, or one of the meeting participants themselves takes on the task. As high quality note taking requires substantial concentration, the latter variant substantially limits the chosen participant's ability to partake in the actual discussion. In addition, it is usually not sufficient that a person takes notes during a meeting, additional time an effort is required after the meeting to transform the notes into a coherent summary text.

It is thus desirable to support the summarization process by automated means in order to reduce the necessary work-load for the creation of a summary. This is not such a new idea: Chapter 3 shows how research on automatic summarization has developed since the late 1950's. The fact that automatic summarization is still a topic of research half a century later indicates that it is a difficult problem. This is hardly surprising if we consider the complexity of the different tasks people perform when creating a summary: the process involves understanding the original contents, deciding what is and what is not important, and then finally putting the important parts into the shape of a coherent text. In the literature, these three tasks are typically referred to as the *interpretation*, *transformation*, and *generation* phase [Endres-Niggemeyer, 1998; Sparck Jones, 1999]. Each of these tasks is difficult for computers to achieve, yet the quality of the final outcome is directly dependent on their respective performance. This is why traditionally a slightly different kind of summary has been favored for document summarization by the research community, so-called *extractive summaries*. In this kind of summaries, parts of the original documents – typically sentences or paragraphs – are extracted and concatenated to constitute the final summary (see Figure 1.1), often leveraging statistical or machine learning techniques to classify the extract-worthiness of the chosen text units.

This approach has been applied to meetings, too [Zechner, 2001; Murray, 2008]. As a source document, they retract to the transcript of a meeting. However, extractive approaches work best when the source document is of a high linguistic quality, as all the material in the final summary originates more or less verbatim from the source. Conversely, if the quality of the source document is already rather poor, e.g., contains typographical errors etc., this will directly influence the quality of the



Source Document

Extractive Summary

To summarize is to reduce in complexity, and hence in length, while retaining some of the essential qualities of the original. This paper focuses on document extracts, a particular kind of computed document summary. Document extracts consisting of roughly 20% of the original can be as

Figure 1.1: Extractive summarization concatenates portions of the source document to create a summary text *(text from Kupiec et al. [1995])*.

generated summary. Meeting transcripts, which record word by word what each of the participants said during the meeting, can in general not be expected to display a high linguistic quality. They differ substantially from carefully drafted documents in that their contents are produced spontaneously. They typically contain ungrammatical and elliptical utterances, speech disfluencies, such as, stuttering and filled pauses, colloquialisms, first-person wording, and other effects that are undesirable for high-quality summary texts. When the transcript is created automatically by an automatic speech recognition system, erroneously detected words are another problem that decrease the overall quality if they end up in the produced summary. Some of these issues can addressed by pre-processing the source and/or post-processing the summary text. But a thorough treatment of these factors requires a deep level of analysis and representation that extractive approaches typically don't pursue. We argue that for this reason the classic extractive approach is not as well-suited for speech-based summarization as it may be for document summarization.

An alternative summarization approach is sometimes referred to as *abstractive summarization*. Unlike extractive summarization, abstractive approaches attempt to mimic the way people write summaries. It uses a symbolic representation for the contents of the source and the summary, and the final summary text is generated from the summary representations rather than extracted directly from the source text. Using Natural Language Generation (NLG) to create the text of the summary makes the generation phase of the abstractive approach more complex, but carries the potential to overcome some of the weaknesses of extractive summarization outlined above. In particular, issues of text cohesion and coherence in the created summary (e.g., inter-sentence relations, dangling anaphors, etc.) can be treated in-

Table 1.1: Comparison of the interpretation (I), transformation (T), and generation (G) phases for abstractive and extractive summarization

	Abstractive Summarization	**Extractive Summarization**
I	Use NLU to create symbolic represenation of contents.	Extract features for every unit of text.
T	Transform source representation into summary representation.	Evaluate units based on features.
G	Use NLG to create text from summary representation.	Extract best units from source text and concatenate them.

dependently of the source document. Also, in contexts where the source consists of spontaneously spoken language, typical speech artifacts, such as ellipses or disfluencies, that are problematic for extractive summaries can be avoided when the summary text is generated.

Table 1.1 shows a comparison of extractive and abstractive summarization with respect to the three abstract phases of summarization. Abstractive summarization approaches use Natural Language Understanding (NLU) methods to arrive at a symbolic representation of the source contents, for instance, using a domain ontology. Such a model allows for automated reasoning methods to transform the source representation into a symbolic representation of the summary contents. Finally, an NLG component transfers this representation into a textual or, less often, multimodal form. Table 1.2 contrasts excerpts of two automatically generated meeting summaries. The summary on the left is an abstractive summary generated by the MEESU system which was developed as part of this thesis. The summary on the right is an extractive summary from [Murray, 2008] which consists of utterances taken from the meeting transcript.

For extractive summarization, the source text is typically represented as a feature vector, at least for approaches based on statistical classification or machine learning, which make up the majority of the published approaches (see Chapter 3). In these methods, a model is derived from a training corpus which is used in the transformation phase to find the most relevant parts of the source. The final generation phase is then quite trivial: the text units that are considered best with respect to the used model are copied and concatenated, usually in their original order.

Table 1.2: Automatically generated meeting summaries for AMI meeting ES2008a: abstractive [MEESU] and extractive [Murray, 2008] (excerpt).

MEESU :	**Extractive**:
The project manager introduces the project to the team. The project manager, the industrial designer and the user-interface designer introduce themselves to the team. The project manager discusses project finances.	PM: My name is Rose Lindgren. PM: Um our agenda today is we are gonna do a little opening PM: then we'll move into acquaintance such as getting to know each other a little bit, including a tool training exercise.

1.1 Abstracting a Meeting Excerpt

Endres-Niggemeyer [1998] provides an elaborate overview of the summarization process both for documents and in everyday communication. In this section we exemplify the summarization of a part of a meeting. We start our analysis with an excerpt of a transcript of a meeting from the AMI Meeting Corpus. This corpus consists of recorded business meetings in which four members of a company collaboratively design a television remote control (see Chapter 4.4). The objective of this analysis is to point out sub-tasks and challenges in summarizing meeting discourse, and to link to the chapters and sections of this thesis that address these points.

Analysis of a Meeting Discourse

The excerpt (Fig. 1.2) begins with two utterances that are of little interest for the overall discourse, *Okay* and *Right.* In this particular case, a summarizer can simply ignore them because it can be assumed that speaker B says them to control the floor and to gain attention for the following utterances. However, in other contexts, the two words may well be important. For instance *okay* could be the answer to an important yes-no question, while *right* could be the confirmation of a significant decision. Such cases should then be recorded in a summary. But in our case, the first real information is uttered in line 3.

The group learns about two points: There is a *project* and the meeting they are currently attending is a kick-off meeting. There area number of implications that can be deduced from these two points, for instance that the group can expect further meetings for the same project. But besides the pure propositional content, we can also conjecture that speaker B has a special role in that project, since she takes the liberty to start the meeting by addressing the group. And in fact, this assumption is confirmed further down in line 8.

1 **B:** Okay

2 **B:** Right

3 **B:** Um well this is the kick-off meeting for our our project .

4 **B:** Um and um

5 **B:** this is just what we're gonna be doing over the next twenty five minutes .

6 **D:** Mm-hmm .

7 **B:** Um so first of all , just to kind of make sure that we all know each other ,

8 **B:** I'm Laura and I'm the project manager .

9 **D:** Great .

10 **B:** Do you want to introduce yourself again ?

11 **A:** Hi , I'm David and I'm supposed to be an industrial designer .

12 **B:** Okay .

13 **D:** And I'm Andrew and I'm uh our marketing

14 **C:** Um I'm Craig and I'm User Interface .

15 **D:** expert .

16 **B:** Great .

17 **B:** Okay . Um

18 **B:** so we're designing a new remote control and um

19 **B:** Oh I have to record who's here actually .

20 **B:** So that's David , Andrew and Craig , isn't it ?

21 **B:** And you all arrived on time .

22 **B:** Um yeah so des uh design a new remote control .

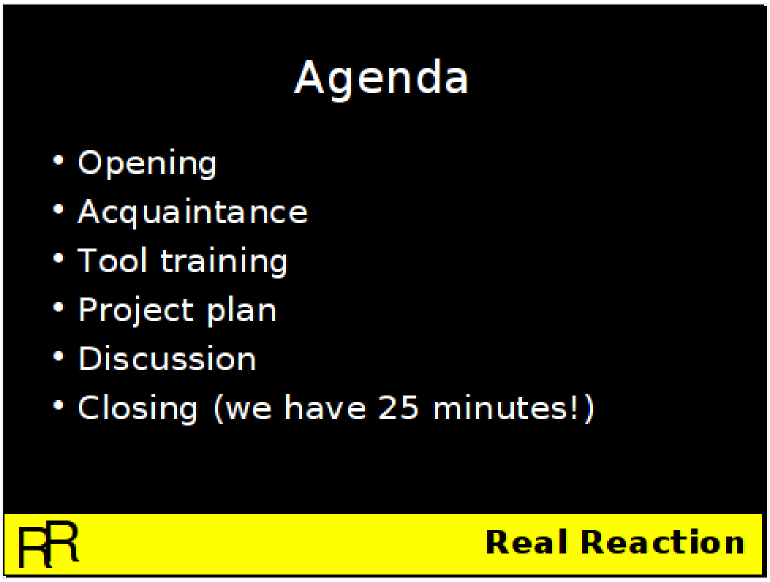
23 **B:** Um , as you can see it's supposed to be original , trendy and user friendly .

24 **B:** Um so that's kind of our our brief , as it were .

Figure 1.2: Excerpt of a manual transcript of meeting ES2002a from the AMI Corpus.

Line 4 shows a typical effect of spontaneous speech: two *um* sounds embrace the word *and.* These sounds are not real words, but speech disfluencies, yet they may serve a function in the discourse, e.g., floor control.

The following line contains a number of different effects. Without further knowledge, it might be difficult to understand because it is not clear what the word *it* refers to. Also, *the next twenty five minutes* is a metaphor for the ongoing meeting which the speaker expects to last 25 minutes. The sentence could be an introductory sentence for a list of things planned during the meeting, but no such list follows. It could also be a sentence that summarizes such a list had it been uttered beforehand, but that is not the case either. Here we witness a fundamental problem in discourse processing, that of resolving anaphora. In this particular case, the video recording of the meeting reveals that B's utterance coincides with the display of an electronic slide showing the meeting's agenda (Figure 1.3). This cannot necessarily be inferred from the transcript and thus an ideal summarizer would be able to resolve multimodal cross-references too.



Agenda

- Opening
- Acquaintance
- Tool training
- Project plan
- Discussion
- Closing (we have 25 minutes!)

RR Real Reaction

Figure 1.3: The agenda slide from the project manager's presentation in the kickoff-meeting in the AMI corpus

From a summarization point of view, the first five lines of the meeting together with the information from line 8 that B is the manager of the project could be summarized as follows: *The project manager showed the agenda of this project kick-off meeting*. This is of course not the only possible way to summarize that part of the discourse. For instance, often times the roles of the participants will be known to the consumers of the summary and so using the participants' names instead of their role may read more naturally.

We observe some typical strategies that are applied when summarizing a discourse such as a meeting transcript, for instance, the deletion of less relevant information (*Okay*, *Right*, *the next twenty five minutes*, etc.), or the construction of a new proposition that does not appear in the original source, but can be inferred from the propositions that do.

Line 6 shows another non-verbal sound. This time, it does not function as a floor control mechanism, but D's *Mm-hmm* sound signals to the main speaker B that D is following along what B is presenting. We call such a signal a *backchannel* (see Section 4.4). Further examples of backchannels can be seen in lines 9, 12, 16, 17. For a summary of this part of the discourse, these particular backchannels are of no consequence.

In line 7 another typical effect of spontaneous speech can be observed, elliptical expressions. The utterance is not a full grammatical sentence because it only consists of an adverbial phrase and a sub-ordinate clause, but lacks a main clause. But we can still gain valuable information from the utterance. The speaker talks about the purpose of something and even though it is not revealed what that something really is, we can with some certainty infer it from the purpose of that sentence: since the result is "knowing each other" it is quite safe to assume that the act itself is that of introducing themselves to each other.

This rhetorical device, making the listeners infer the intended meaning instead of saying it outright, is not uncommon, but it may fail, e.g., if the expected inference is too complex or unexpected under the current circumstances. In fact, when we look at line 10, we see that B encourages the other group members again to introduce themselves, this time more directly, even after she already set a good example and introduced herself in line 8. This could be read as an indication that B's attempt to establish an round robin introduction was not successful. At the same time, it could also be understood as a confirmation that the interpretation of the ellipsis in line 7 was to the point.

The participants introduce themselves in lines 8, 11, 13, and 14 respectively. The patterns happen to be quite similar, they all say their first name and their profession. The fact that A says in line 11 that he is *supposed* to be an industrial designer is an artifact from the way the AMI Corpus was created. The participants of each meeting do not really work together in a company that produces remote controls. Rather, they reenact this scenario which they have been briefed about before the meeting. By saying "supposed to", participant A falls out of role in line 11, a fact which we ignore for the purpose of this exercise.

Up to line 17, we could thus extend our textual summary of the meeting with the sentence *The participants introduced themselves to each other by name and role in the project.*. We arrive at such a sentence through a third summarization strategy by which we abstract a sequence of similar utterances and only verbalize the common features, here the fact that all participants perform an introduction of themselves to the group.

Line 18 contains a topic shift. So far we have seen two topics in the discourse (besides the opening remark of line 3), *presentation of the meeting agenda* and *introduction of the meeting participants*. These topics occurred in a linear order, i.e., the meeting started with one topic and finished it before moving to the next topic which was again finished before a new topic comes up now: *an overview of the project task*.

B declares in line 18 that this task consists of designing a new remote control.

Topic shifts are interesting in so far as they provide a structuring of the meeting. The topic segments could be used as the basic units of summarization as we have exercised above by generating one sentence for each of the topics so far. But the following line, 19, marks a departure from the linear execution of meeting topics. Before the discourse topic that started in line 18 has been finished, a new topic barges in, *formal recording of meeting participation*. Lines 19–21 are concerned with that topic which overlays the previous, unfinished topic. The new topic itself gets finished correctly with line 21, and we see that in line 22 the previous topic gets restored. [1] The overlapping topic could be summarized as *The project manager recorded the attendance of the participants* but we argue that such formalities are typically not relevant enough for inclusion on a summary and should probably be left out in most cases. That does not mean that a list of participants should not be included in a summary, in fact, such information could well be found e.g. in a formal header together with other data such as the date and time of the meeting. But the fact that the project manager wrote down a list of participants and their arrival times does not necessarily have to be included.

Line 22 picks up the task of the project again, the used speech material is very similar to that of line 18. The following line references the slide presentation again when B says *as you can see*. We learn that certain attributes are expected of the new remote control the group is to design. Line 24 closes the overview of the upcoming project that B gives to the group. That part of the discuss could thus be summarized as: *The project objective is the design of an original, trendy and user-friendly remote control.*

A complete summary of the example excerpt, retaining the original order of the discourse, could thus be written as follows:

> *The project manager showed the agenda of this project kick-off meeting. Then the participants introduced themselves to each other by name and role in the project. The project objective is the design of an original, trendy and user-friendly remote control.*

However, in the abstract that is part of the official corpus annotation, the annotator decided to leave out the parts about the meeting agenda and about the introduction of the meeting participants. The abstract begins instead with the briefing that participant *B* starts giving to the group toward the bottom of our excerpt:

> *The project manager introduced the upcoming project to the team members.*

[1] This is reminiscent of a *stack* data-structure where beginning a new topic corresponds to a *push* and finishing a topic corresponds to a *pop*, but in general a stack is not an apt model of topic shifts because the focus of discourse can move more freely than can be modeled by *push* and *pop* operations.

This exemplifies that summaries can be written at varying levels of detail. The official abstract in the corpus is less detailed than our own analysis in that it leaves out more information. For instance, while we re-iterate in our summary what the project is about, the official summary only mentions that the project manager talks about that topic, but not what she actually says.

The most compelling reason to choose less detail for a summary is to reduce the size of the summary. This is not surprising as the main rationale for summarization is to condense the source to a smaller version. It is up to the summarizer to decide on the level of detail. The level of detail in the summary does not necessarily have to be distributed evenly over the underlying discourse. If a specific user need is known, a summarizer might decide to summarize one part of the source in a rather abstract way while incorporating more detail in the part that is known to be the most valuable to the reader.

The informational content of a summary is typically dependent on the summary's length. The shorter a summary, the more difficult it becomes to make it informative. Ideally, a summary is as short as possible and as informative as possible (see Figure 1.4), but because of the interdependence of length and informativeness this is not easily achievable.

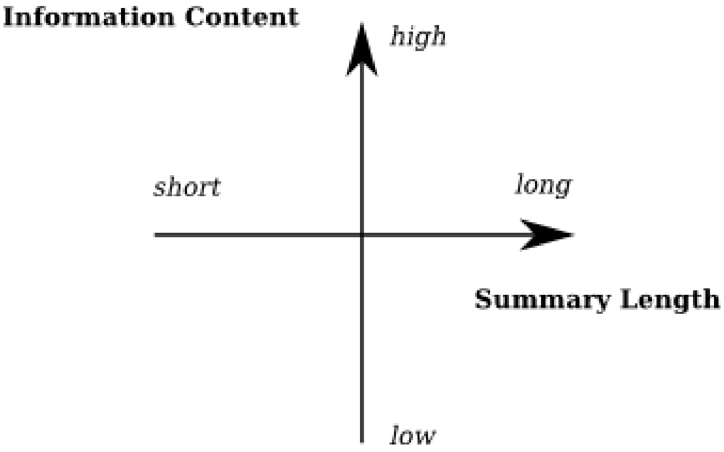


Figure 1.4: Informativeness and length of a summary.

Discussion

The analysis of the excerpt has provided us with a number of interesting insights about the process of summarizing a meeting. In particular we are able to identify some challenges for automatic interpretation of a transcript, its transformation into summary contents and even for the generation of these contents into English text. This section reprises and discusses the findings from the analysis in the previous section and derives from it guidelines for automatic summarizers. We also refer to the chapters and sections of this thesis in which the discussed points are addressed in greater detail.

Context Dependency

Utterances can take different meanings depending on the state of discourse at which they are uttered. For instance, we note that the word “okay” can serve as a confirmation or a backchannel or have even further meanings. It is thus important for an automatic summarizer not to interpret utterances in isolation but take the current state of discourse into account.

Propositional Content

An abstractive summarizer should have the means to understand propositions expressed in speaker utterances. Content abstraction means to reason about what is being said in a meeting discourse, infer implications and combine statements made at different points into a coherent representation.

Such operations cannot be done without representing the propositional content of the meeting discourse.

World Knowledge

In order to understand the discourse between the meeting participants, a certain knowledge about the world must be available to the summarizer. For instance, to understand line 3 of the above excerpt, a summarizer must know what is meant by *kick-off meeting* which refers to the first meeting in a project that marks the beginning of that project. This involves many different kinds of knowledge, among them:

Ontological knowledge

The concepts *project*, *meeting*, etc. must mean something to the summarizer if he or she (or it, in case of an automatic summarizer) is to understand what is being talked about. An ontological model should make clear what the properties of these concepts are, e.g., that they are *perdurants* [Masolo et al., 2003] that have some fundamental categorical differences in comparison to such concepts as *remote control*, *agenda*, or *meeting participant*. It should also model what relations can hold between instances of such concepts, e.g., that a meeting can be a *part* of a project (cf. 4.4).

Script knowledge

To understand how the kick-off meeting relates to the project, the summarizer must have an understanding of the way business projects are typically executed, i.e., that the work in a project contains of individual and/or group work of the members, that there are occasional or regular meetings, that communication happens also outside the meetings, that reports get written, etc. Such procedural knowledge is called *script knowledge* by Schank and Abelson [1977]. It is that kind of knowledge that lets us infer that the ellipsis *just to [...] make sure that we all know each other* refers to an introduction of the speakers to each other, as the proposition *the members know each other* is an outcome of a proposed *Getting_acquainted* script.

Behavioral content

A special kind of world knowledge is that about typical or expected behaviors of people from which certain conclusions about their roles and status can be drawn. For instance, the fact that B is the project manager is stated explicitly in line 8, however, even without that particular utterance, we could have hypothesized that she might be from the fact she took initiative and started the meeting. By convention, in business meetings of this size it is usual that the participant with the highest status takes the lead.

Such behavioral knowledge can thus be used to infer certain properties that are never made explicit in the discourse. It is, however, often subject to cultural, habitual and situational constraints.

Inference

Knowing which further propositions can be inferred from a given set of propositions can be practical for a summarizer in a number of different ways.

As mentioned before, inferring further propositional knowledge in addition to the propositions that can be extracted directly from the discourse can help to fill an otherwise sketchy account of the meeting contents. This can be paramount for an abstractive summary that has to take into account the full picture of the things discussed. We have already seen that human communication tends to utilize a technique in which propositions are expressed indirectly through logic entailment.

At the same time, a summarizer can make use of a similar technique if it has as its command a model of what certain propositions entail. For instance if two propositions are considered relevant by a summarizer, but one entails the other one, it could suffice to only add the first one to the summary model. In turn, if a set of propositions entails another proposition, it might be sufficient to only include the entailed proposition in the summary. This kind of reasoning build the basis for two of the *macro-rules* described below.

Linguistic knowledge

Since the basis for our analysis of a meeting excerpt has been the transcript of that excerpt, it is clear that some linguistic knowledge is required to extract the aforementioned propositions. It is necessary for an automatic summarizer to have means to map the tokens of a transcripts, i.e. the words and transcribed sounds, to the knowledge structures used for the representations of propositions.

This requires a method to represent the meaning of words in the context they appear in, which amounts to the classic field of linguistic semantics. Again, semantics is only a building block in understanding human discourse. A complete model of human communication would also have to encompass fields such as pragmatics, the study of metaphors, metonymy, to name a few.

Reference resolution

One particular sub-field in the process of interpreting the linguistic material found in a meeting transcript is *reference resolution.* Classically, this means the resolution of linguistic co-references, anaphoric or cataphoric, i.e. figuring out which words refer to the same entity. As we have seen above, in a meeting setting not all information is found within the transcript, and so multimodal cross-references would ideally have to be resolved, too.

In addition to finding out which words mean the same entity, a summarizer must also have a means to find out *which* entities these are. Assuming that our model of world knowledge (see above) has a representation of real-world entities, the task thus is to map between words and the model of concrete entities.

Speech irregularities

One stepping stone on our way to interpret the discourse of a meeting is the fact that the meeting participants speak spontaneously and therefore include in their speech artifacts typically not known in, say, written language. The utterances in a conversation are more often than not ungrammatical and an automatic summarization system has to be able to deal with this fact.

The above excerpt already exemplified typical such effects. Disfluent speech could be observed in form of non-word sounds (*um*, *mm-hmm*), repetitions of words (*our our*), stuttering (*des design*) and fill-phrases (just to *kind of* make sure).

Likewise, ellipses i.e. omissions of clauses or one or more words, that are not only produced involuntarily as a specific kind of speech disfluency, but also used deliberately as a rhetorical device, result in ungrammatical utterances.

Topic shifts

Unless in very short and focused meetings, we can expect the discourse to cover multiple topics. A summary of such a meeting should thus be capable of distinguish between the different foci of discussions. Topic boundaries provide a natural structuring of a meeting. If these boundaries are ignored, different topics may get mixed up in the analysis of the discourse which may lead to wrong inferences and may affect the subsequent relevance assessment in a negative way.

Relevance

Detecting what is relevant and what is not in a discourse is arguably at the heart of any summarization process. The point of summarizing is to concentrate on what is relevant in the source, and only include these parts in the summary, leaving out less important parts. Relevance assessment is not an absolute task, it is dependent, among other things, on the source, the summary producer, the summary consumer, a potential specific information need, etc. For instance, the official abstractive summary of the above meeting excerpt available in the AMI corpus does not rule the presentation of the meeting agenda nor the introductory round relevant enough for inclusion.

Traditional measures for relevance in summarization look at the distribution of content over the discourse: concepts that appear often are assumed to be important. In written documents, certain locations in the document have been shown to correlate with more important contents, but for meetings that develop spontaneously and collaboratively, it is not clear whether such findings translate. The same is true for a third hint, the use of typical key phrases that in documents serve as a good indication for important material.

A summarizer may decide to include in the summary certain structures independent of the above clues, using domain knowledge. For instance, in business meetings a summarization strategy could be to include all recognized decisions, even if other measures deem them not relevant enough.

Macro-rules

Once a discourse has been fully understood, the question is how to abstract it into a summary. We have seen above three basic techniques: the deletion of irrelevant parts (recording of the participant's names and arrival times), the abstraction of similar parts to a more general proposition entailed by these parts (*they introduced themselves* instead of *B introduced herself, A introduced himself, C introduced himself, D introduced himself,* and the construction of a new proposition that is conventionally known to consist of the propositions for which actual evidence is found in the meeting discourse (*they introduced themselves* instead of *they said their name and their roles in the project*).

We call such operators *macro-rules* (see Section 3.2). They are different to the general inferences outlined above in that they are not part of the discourse understanding phase, but come conceptually later. Macro-rules are applied on top of the results of discourse understanding, making use of the propositions and representations that have already been extracted or inferred. They perform the actual content reduction which results in a representation of the contents for the final summary.

Non-linearity

In general a discourse such as a meeting is not a linear sequence of topics, especially when discussions are involved. Previous discussions get revisited, perhaps multiple times, which poses a difficulty for discourse understanding. If a topic shift is detected, the next question is whether this is the start of a completely new topic or whether it is a reprise of a previous topic.

Depending on whether the focus of summarization is on results or on the chronological progression, a relevant topic that reappears multiple times in the course of a meeting could be included in the abstract once in its entirety, or multiple times similar to the source.

Order of summary content

Summarizing may involve re-ordering and re-structuring information found in the source. One such case is topic shifts, discussed above. The question in which order overlapping topics should be presented in a summary is in fact a question that could be asked for all meeting topics, not just overlapping ones. Different strategies are conceivable, such as, the original order from the meeting, or by decreasing relevance so that the most important appear toward the beginning of a summary. Reusing the order in which different topics were discussed in the meeting has the advantage that it is straight-forward for the participants themselves to find information in the summary.

1.2 Meeting-related Research Projects

A number of international research projects have concerned themselves in the past with meetings and the role of computer-mediated support in group interactions. The "ICSI Meeting Project"[2] [Morgan et al., 2001] was one of the first large-scale projects dedicated to research in that area. In the years 2000–2002, 75 informal, natural meetings were recorded at the International Computer Science Institute (ICSI), Berkeley, CA. [Janin et al., 2004] using different variations of microphone setups simultaneously. As one of the main interests of the involved researchers was in auto-

[2] alternatively called "ICSI Meeting Recorder Project" at times

matic speech recognition, only the audio tracks but no videos of the meetings were recorded. The project also worked on other automatic means for analyzing group interactions, such as segmentation of the discourse, detection of hot spots [Wrede and Shriberg, 2003], and the classification of dialog acts [Shriberg et al., 2004] (see also chapter 5.3. The records of the meetings together with manual transcripts and some annotations have been released as the "ICSI Meeting Corpus" which is available through the Linguistic Data Consortium [Janin et al., 2003; Shriberg et al., 2004].

The "M4" project (*Multimodal Meeting Manager*) was a European research project, funded by the EU under its IST Programme, in the years 2002–2005. The project performed research on structuring, browsing, and querying meetings that were previously recorded in an instrumented meeting room and analyzed automatically. A key difference in comparison to the ICSI Meeting Project was the inclusion of video channels in the meeting recordings, which allowed for multimodal analyses of the recordings.

A third, more recent project was the AMI project with it successor AMIDA. We are going to describe this project in greater detail in Section 4.4, especially the corpus it produced, since this corpus provides the basis for the research presented in this thesis.

1.3 Research Questions

This thesis argues that for the reasons outlined above, an abstractive approach is better suited for meeting summarization than an extractive approach. This claim will be substantiated by an operational model for abstractive meeting summarization, developed in subsequent chapters, and its implementation in form of a computer program called MEESU. In doing so, this thesis addresses and provides answers to the following research questions:

1. **What is a viable design for an abstractive meeting summarization system?**

   The overall question that this thesis addresses is how an abstractive meeting summary can be generated automatically that approaches the quality of hand-written summaries. More specifically, we aim to identify the sub-steps that allow a machine to produce such a summary, i.e., the required steps that start with a given meeting and ends with the final textual document. We also study how and in which ways the components implementing these steps interact with each other, and what kind of encoded knowledge they draw upon.

2. **Can such a design be implemented using readily available knowledge sources for language processing?**

Previous abstractive approaches have made use of manually crafted knowledge representations, such as ontologies. Creating and maintaining such representations can be tedious work, and yet they are limited to specific application domains. Existing extractive approaches on the other hand tend not to use semantic representations at all, but favor directly observable features. This allows for relative domain independence, but comes at the cost of a reduced textual quality of the produced summary text. An extractive summary depends directly on the quality of the meeting transcript, which is often low due to the spontaneous nature of the conversation. In addition, extracting sentences from different part of the meeting can result in coherence problems. We argue that *Frame Semantics* can be used as the basis of a suitable representation formalism for abstractive meeting summarization.

3. **How can a meeting transcript be mapped onto such a representation?**

   A summarization system that uses a symbolic representation for the contents of a meeting must be able to automatically create such a representation for a given meeting recording. This involves recording a meeting digitally so that it becomes accessible to a computer, as well as different levels of processing, from low-level signal processing via the extraction of lexical information, to higher level semantic interpretation.

4. **Can insights from cognitive science be leveraged as constraints for deriving the contents of the meeting abstract?**

   An ideal summarization system would produce abstracts en par with handwritten summaries. Such an outcome cannot be expected given the current state of the art of component technologies. It is fair to ask though whether the same processes that people use in summarization can be implemented in an abstractive system too. For that we study findings from cognitive science that have concerned themselves with human summarization and propose an algorithm to implement them on top of the employed meeting representation.

5. **How can a content representation based on Frame Semantics be verbalized as text?**

   Many previous publications on summarization do not concern themselves with the production of an actual textual presentation of the generated summary. However, for a complete system, the generation of the final summary text has to be taken seriously. Natural language generation is a distinct research field in its own right. However, to our knowledge there is no existing method to generate text directly from a Frame Semantic representation.

6. **How can the usefulness of a meeting summary be measured?**

The evaluation of automatically generated summaries has been a challenge to the research community since the field was first established. A number of different *intrinsic* methods have been proposed in the past, for some of which varying correlation with human judgment has been observed. Eventually, however, what we are really interested in is the usefulness of generated summaries for realistic, every-day tasks. This calls for a reusable framework for an *extrinsic* summary evaluation in which the performance during a selection of such tasks can be measured and compared across different summarization systems.

1.4 Thesis Overview

This thesis makes contributions to subfields of information extraction, artificial intelligence and computational linguistics. Its objective is to address the challenges listed in the previous section and develop approaches to meet each of them in the context of automatic abstractive summarization of meetings. Because of the complexity of this task, it would be somewhat unrealistic to expect a perfect outcome, thus we are interested in studying which factors influence the quality of the generated summaries. It should also be noted though that measuring summarization quality is a difficult research question in its own right.

In order to address these questions, we first introduce the necessary terminology in the next chapter where we also present the European research projects AMI and AMIDA. A particular focus is put on the AMI Meeting Corpus which has been the basis for the experiments that were carried out for this thesis.

In Chapter 3 we then study previous research related to our task. We will find that abstractive meeting summarization is a niche task in multiple respects. For one, the number of publications dedicated to an extractive approach far outweighs the available literature on abstractive approaches, but we will examine extraction ideas to see whether and how they could be adapted to abstractive summarization. On a similar note, the literature in the field is clearly dominated by work on document summarization, although speech and multimodal summarization has slowly been gaining ground in the last twenty years.

Chapter 5 describes our own approach to abstractive meeting summarization which follows the *Interpretation, Transformation, Generation* model. Chapters 5.4–5.6 detail out each of these steps separately.

We evaluate our model in chapter 6. Here, we argue that traditional metrics for the evaluation of generated summaries are less well-suited for the meeting domain. As an alternative, we describe a novel framework for the extrinsic evaluation of meeting summaries.

Finally, chapter 7 discusses critically what we have achieved in this thesis. It readdresses the research questions and reiterates our contributions. Abstractive meeting summarization is a relatively young field of research the approach presented here is novel. As such, it provides the opportunity for quite a few future directions. A discussion of theses closes this thesis.

Chapter 2

Terminology and Classification

2.1 Introduction

This chapter introduces and discusses some of the terminology used throughout this thesis. We motivate the usage of specific terms through discussion of alternatives or by juxtaposition to related terms. To start off, we show that although we possess a strong intuition about the meaning of the word "summary", it is not trivial to give an exact definition for the term. This is exemplified by a discussion of definitions suggested by the American National Standards Institute (ANSI) in Section 2.2 where we detail a number of shortcomings.

After that, we give our own definitions of an abstractive summary and related terms in Sections 2.3 and 2.4.

Independent of a distinction of form, e.g. extractive vs. abstractive, summaries can be further categorized according to a number of different dimensions, which this chapter presents in Sections 2.5–2.9. In particular, Spärck Jones' classification scheme is discussed (Section 2.10). The chapter closes with an application of the scheme to the task at hand. Here, we suggest a categorization of abstractive meeting summarization in terms of input, purpose and output factors.

2.2 Existing Definitions

If our ultimate goal is the automatic generation of meeting summaries, the first question we have to ask ourselves is: "what is a summary?". What looks like a simple question is in fact more difficult than it seems. In the early days of research on automatic summarization, the terms "summary" and "abstract" were used synonymously (cf. e.g. [Luhn, 1958])–not so in the ANSI[1] *Guidelines for Abstracts* [ANSI-96] where the following distinctions are made:

Summary A brief restatement within a document (usually at the end) of its salient findings and conclusions intended to complete the orientation of a reader who has studied the preceding text.

[1] American National Standards Institute

Abstract A brief and objective representation of a document or an oral representation.

Extract One or more portions of a document selected to represent the whole.

All three of these definitions rely on a fourth, that of a *document* which ANSI defines as well. The definition includes printed and non-printed material as well as three-dimensional objects or realia such as museum objects and specimen. Unfortunately, this must be considered a vague definition with the only unity criterion for these different items being that they are "amenable to abstracting" which makes the definition circular. Also, it is questionable whether a meeting would fall under the ANSI definition of a document at all. A recording of a meeting would classify as a document according to the definition, but summarization of life events do not seem to be covered by the ANSI definition.

Even if we assume for the moment that a meeting could be taken for a document in the ANSI sense then the notion of "summary" still only refers to in-meeting summaries, i.e., summaries that the meeting participants themselves create as part of the meeting discourse, e.g., in the form of a wrap-up at the end of meeting. For our purpose, this notion is too narrow as we aim for *external* summary generation after the meeting is over. The term "abstract" in the ANSI sense is closer to this idea, but again we face vague descriptions, namely for the terms "brief" and "objective". In contrast, the definition of "extract" seems to be straight-forward and clear, although it is tied to a particular purpose which is questionable. A summary may well serve many different purposes (cf. section 2.6).

Brevity is of course one of the key aspects of a summary. It is probably the main motivation for summaries to begin with: a summary's main purpose is to free the user from having to access the full length of the source document. This does not mean, however, that a summary is generally intended to replace the source document (s. discussion of "informative" and "indicative" below), a summary may in fact be incentive for the reader to also read the full document.

To have a measure of brevity, the term *compression* is often used as the ratio between the length of summary and that of the source [Mani, 2001]. Alternatively, some authors use *compression* to stand for 1 minus that ratio. However, the length of a (text) document is difficult to measure: is it the number of pages? The number of lines or words? The time required to read it? Other but similar questions could be asked for dynamic media documents: is the length of a video recording the number of seconds it lasts? Think of a recorded meeting: when is the exact start of it? When is the exact ending? Does a playback at 1.5 of the original speed already constitute a summary [Tucker and Whittaker, 2008, cf.]? And how can we measure the compression for cross-media summaries, such as a textual summary of a meeting?

Answers to questions like these are not given by the definitions of ANSI. All in all, the relative vagueness and unnatural restrictions they display are motivation enough to look for alternative definitions that bare the potential of being more suitable for the task. One reason for the drawbacks of the ANSI definitions could be that they are tailored toward human abstractors in the first place. We shall therefore turn toward the literature in the field of automatic summarization research and examine the notions of summary etc. used therein.

Sparck Jones [1999] specifies the term summary *en passant* in her definition of *summarization* as "a reductive transformation of source text to summary text through content reduction by selection and/or generalisation on what is important in the source." This definition is obviously targeted at *textual* summarization, Spärck Jones assumes that both the source and the summary will be of textual form. This is not necessarily so, for instance, Maybury [1995] generates summaries of event data, VITRA-SOCCER [Herzog and Wazinski, 1994] summaries of video scenes. The limitation to the text modality seems unnecessary, at least for the source. For the result of the summarization process it is perhaps less clear whether it has to be of textual form to be considered a "summary". The question is apparently more difficult: on the one hand, non-textual summaries are certainly widely used – for example, news broadcasts extract the most important scenes of a recording when reporting on a soccer game (goal scenes, red cards, etc.). On the other hand, an extract of the most important data from a long series of event data would in general not be considered a summary–although a textual version of the same data extract might be. Yet we cannot conclude that low-level event data are inapt as a medium for summaries per se, as we can see by what perhaps is the most compact summary of a soccer game: the names of the teams plus the number of goals each team scored, e.g., "England 3 : 2 Germany". These very basic data, when printed in a newspaper, are vital summaries for millions of soccer fans. All in all, we conclude that a restriction to a particular medium should not be a requirement for a summary.

According to Spärck Jones' definition, what sets summaries apart from other documents derived from a source, such as, e.g., an index or a list of keywords, is the focus on "what is important in the source". This notion is not unproblematic as "importance" is not clearly defined itself. One could define the importance of a document with respect to a specific information need – in such a setting it is often synonymously referred to as "relevance". Such an approach was taken, for instance, by the document understanding conference (DUC) in 2005 when they introduced a track on *query-based summarization* in which a user query represents a specific information need. This interpretation makes "importance" a *context-dependent* feature and consequently what is important in a document in one situation might be considered unimportant in another situation. Still, summaries are typically produced for future use and thus the situation or context in which they will be consulted is unknown at the time of production. In that light, if importance is considered be a key factor for a summary, how could the authors of a summary claim that they are creating a summary, unless they know with certainty that the "right" context for her summary will become reality and some point in the future?

For summaries that are not generated "on demand" given a specific query, one way to deal with the notion of importance is to somehow average over all *expected* queries that this document could be relevant for thereby relaxing the creator's dilemma which we sketched above. Of course, such an approach must be an approximation because the set of all queries cannot be known in advance. It is up to the summarizer, man or machine, to develop an idea of a "most typical" query that ideally would not possess too much of a bias into any particular direction and to which a summary should provide an answer. We take it that an idea similar to this one is what is meant by the term "objective" in the ANSI definition of "abstract". Again, we have to deal with a vague notion which is why a good and general definition of "summary" should avoid such a notion altogether.

Spärck Jones' definition also contains some hints of how we may arrive at a briefer, reduced version of the source. She names two possibilities, selection and generalization which might be combined or used independently. However, Endres-Niggemeyer [1998, 54f] describes at least one more possible strategy to information reduction, called *construction*: "Given a sequence of propositions, replace it by a proposition that is entailed by the joint set of propositions of the sequence." (cf. Section 1.1). Construction was not anticipated by Spärck Jones. This shows that including the *means* by which to achieve a result (the summary) into the definition of the process can result in undesired limitations. We therefore propose to leave out any allusion to means from the definition.

To this point, we have shown a number of difficulties that arise when trying to define precisely the seemingly intuitive terms "summary", "abstract", and "extract". But we have still not arrived at a useful definition of the above terms.

Therefore we will propose our own definitions below. One lesson learned from the above excursion should be that we cannot expect that our own definitions are more useful or contain less problematic aspects than any of the notions above. Nevertheless we introduce the following terms as a basis for subsequent usage in the following chapters.

2.3 Content, Documents and Summaries

Summaries are necessarily summaries some contents which we call *source*. For this work, the contents we are mostly interested in are meetings. However, we do not work directly on the meeting itself, but on a transcript of what the meeting participants discuss. A transcript takes the role of textual documents in text summarization. In fact, we can consider the transcript to be a particular kind of document. Since the document is the starting point for all summarization, we introduce the following definition.

Definition 1:
A *document* is a physical serialization of some contents.

This is a quite general definition. It is clear that any document is realized in at least one medium, e.g., text or video. Meeting transcripts are typically textual, but with the availability of video and audio recording facilities, we could also imagine a multimedia transcript of a meeting. For now, this is beyond the scope of this thesis, though, where we deal exclusively with textual transcripts.

A summary is typically also encoded as a textual document. However, the concrete medium is not crucial for this, as a summary could also be reported orally, for example. Rather, we have to understand a summary in terms of *contents*. A summary should be seen as a *subset of the source contents*.

Not all such fractions of some given contents would be considered a good summary thereof. Ideally, only the most *relevant* parts of the source contents should be incorporated into a summary. But if the important aspect was only relevance, the original source would have to be considered the ideal summary of itself since it contains *all* contents and thus implicitly all relevant contents too. The defining characteristic of a summary is that it also does not contain many irrelevant contents, and thus allowing faster access to the relevant ones.

However, this thought presupposes the existence of a single valid relevance measure which does not exist, a measure of relevance only makes sense in the context of specific information needs. In other words, the same subset of some given contents may in one situation be considered a good summary–namely when it helps in fulfilling a particular information need–while in others it won't.

Definition 2:
Given two contents S and s, we call s a *summary* of S if there is a set of information needs I so that all $i \in I$ that can be met by S can also be met by s, and in considerably less time.

Note that the contents to summarize may be present in the form of a document. A summary itself may be serialized as a summary document. When clear from context which one we refer to, we may call both the summary and the summary document "summary".

2.4 Extractive and Abstractive Summaries

Since summary are derived work that draw contents from a source document, it seems straight-forward to have a summary document actually *consist* of original material from the source. For instance, if the source is a newspaper article, a summary may consist of sentences taken verbatim from that article. As a matter of fact, one could argue that the title, extracted from the article, could serve as a minimal summary for an article. In fact, some news sites on the World Wide Web present their articles as lists of titles with links to the full content.

To create longer and more verbose summaries in this style however, one would rather take the most representative sentences from the article and a concatenation of them as the summary. We call such a kind of summary document an “extractive” summary document, or short “extract”.

Definition 3:
Given some source contents S, a summary s of S, a document D that serializes S and a document d that serializes s, we call d an *extractive summary document* if $d \in D$.

Extractive summaries have a number of desirable properties. First, they are very close to the source in their wording, literally re-using the original content. They thus circumvent the creation of new text which would otherwise be subject to a potentially biased interpretation by the summarizer. The contents in the summary are quotations from the source. Extracts are also quite easy to create in a technical sense - “writing” a summary becomes a mere selection process. For textual representations, this could be as simple as copy and paste within a word processor, or, in a specialized extraction system, become a matter of a single mouse click for each sentence to be selected. Given a suitable segmentation and easy access to the segments, the latter process is not limited to textual summaries but could be conceivable even for dynamic media such as audio and video. Even if the segments have to be determined, too, by the summarizer, the overall process of creating an extractive summary is still relatively simple.

In contrast to extractive summaries, we speak of *abstractive* summaries when the contents of the summaries are generated anew by the summarizer instead of copied over from the source. This notion is perhaps more in line with the traditional idea of crafting a summary: the summarizer first “understands” the source and then writes down (in case of a textual summary) *in their own words* what the source is about. Of course, since the summary and the source document are both about the same contents, there may be some overlap in the linguistic realization.

Definition 4:
A summary document of a summary s of S is called *abstractive* when it consists in large parts of material *not* taken of any serializations D of S, but created independently.

Abstractive summarization allows for a higher degree of freedom and control of what will end up in the summary, as the summarizer is not limited to quoting material from the source. In principal, this allows for higher degrees of abstraction (hence the name) and compression. This may on the other hand call for more skills on the summarizer's side.

While extractive summaries obviously have to be authored in at least some of the modalities that the source uses, abstractive summaries are less dependent on the source modalities. If the source document is, e.g., a video recording it is fairly difficult to create a textual extractive summary from that. In order to be able to do that at all, an addition transfer to a textual representation of the source is necessary. This is, of course, not the case for abstractive summarization since here, the summary is always created from scratch in any case.

In everyday life, summarization might be of a *hybrid* form between *extractive* and *abstractive*. For instance, Endres-Niggemeyer [1998] reports of professional summarizers who combine in their work parts of the original document with own contents when summarizing scientific articles.

2.5 Progress- and Result-oriented Summaries

Depending on their indented use, summaries often fall into one of two categories: *progress-oriented* or *result-oriented*.

Progress-oriented summaries summarize the structure of the source. For dynamic sources, such as, audio/video documents, this means that such summaries reproduce the temporal structure of the source by describing the sequence of relevant events. For static sources, such as text documents, a progress-oriented summary is one that adheres to the logical structure of the source and summarizes the contents of the document in the same order in which they are present in the source.

In contrast, result-oriented summaries do not report the order of events of a source but list their outcomes. The order of the outcomes could be inspired by the order of occurrence in the source, but that is not a requirement. The order could also be determined by their absolute importance or by a natural ordering inherent in a given domain. For instance, result-oriented summaries for dialogs about planning a meeting might follow a fix structure: *(1) date, (2) time, (3) location,...* of the actual times at which these points were actually decided during the dialog.

It should be noted that a summary must not necessarily be considered strictly either progress- or result-oriented. Often times, a progress-oriented summary will still name some results and a result-oriented summary may reflect the order of the results as they occurred in the source. As we will see below (see chapter 4.4), the manual summaries in the AMI meeting corpus follow a pre-defined structure that contains both, a progress-oriented section of the course of a meeting as well as the major results of the meeting, with the latter section further divided into three subsections.

2.6 Informative and Indicative Summaries

The difference between *indicative* and *informative* summaries [Borko and Bernier, 1975] is that of the relation of the summary to its source. What role does the summary take, what is it aiming to achieve in comparison with the source: should it be a full replacement of the source, freeing the reader from ever having to consult the original source, or should it just give enough hints to the readers for them to make a well-founded decision whether it would be worth consulting the (longer) source.

The first of the two cases is called an *informative* summary and the second an *indicative* summary. While at a first glance, an indicative summary may seem the easier one to produce, as it “only” needs to give a rough overview of the source, this type of summary can nevertheless be very useful. In what is perhaps the most widely used application of automatic summarization in everyday life—web search—results to a query are typically presented as links to the source documents together with a short indicative extract. This extract often times is the sole basis for the user on which to judge relevance to their information need and thus illustrates impressively the benefit of indicative summaries even as short as two or three lines.

In practical application, the distinction of whether a summary is indicative or informative is likely to be less clear: an informative summary might still have the reader consult the source document if more detail is required than the summary can provide–thus rendering it effectively *indicative*–while an originally indicative summary may sometimes already contain just the information the reader was looking for and a retrieval of the summary’s source is unnecessary.

As mentioned above, whether a summary is considered indicative or informative lies in the eye of the consumers and the particular situation in which they are using the summary, rather than in the summary authors’ hand.

2.7 Generic and Query-Specific Summaries

Another point that we alluded to in the discussion at the beginning of this chapter is whether a summary is created to meet a specific information need or whether it should present a more general view of the source contents.

We understand such a specific information need as formulated through a kind of query. Again, web search could serve as an example familiar to the reader where the so-called “snippets” that are returned together with the title of a web page when thought to be relevant to a query the user typed before. These snippets are a few lines extracted from the web page in question that should show to the user of the search engine in which way the returned page is relevant for her query. Snippets are not pre-stored, but generated on-the-fly as *query-specific* summaries.

In contrast, summaries may be created without a particular information need in mind. In such a case, the summary creator typically tries to present what is considered the most important content as unbiased as possible. We call this a *generic* summary.

2.8 Single and Multi-Source Summaries

So far, we were able to use the terms *document* and *contents* almost synonymously, but it is conceivable that the contents to summarize are actually spread out over multiple documents. In that case we speak of a *multi-source* summary.

Special attention has to be paid to potential overlap of contents realized in the different sources. The identification of such overlap can prove to be a difficult talk for automatic systems, as can the detection of contradictory information. A follow-up question than is how to deal with contradiction in case they are detected.

A special case of multi-source documentation could be seen in situations where input to a summarizer is presented in different modalities. This is of particular interest for meeting summaries, as the available information could be distributed over different channels (video and audio recordings, external sensor data, etc.)

The traditional case for summarization, however, is that of summarizing a *single* document.

2.9 Further Dimensions

Some researches have postulated further distinctions and although not all of them are applicable to meeting summarization, we briefly list some of them here.

Critical and aggregative Some researchers (e.g. Lancaster [1998]; Mani [2001]) add further values to the *indicative/informative* dimension. A summary that not only reproduces contents from the source but adds own comments is referred to as *critical* or *critical evaluative*. A fourth kind of summary is called *aggregative* when it is based on multiple sources which it deliberately sets out in relation to each other.

Background and *What-is-new?* summaries Depending on the familiarity of the summary consumer with the topic at hand, it may or may not be necessary to provide background information. A frequent consumer may prefer to only be presented with an update on a otherwise known background.

Modalities of summaries A summary may combine different source modalities which could be present in a single source document or in multiple different sources. The summary itself could use the same modality as the source or transfer one modality to another one, e.g. text to audio (cross-modal summary).

2.10 Classification of Meeting Summarization

To organize such dimensions with which we can classify different types of summaries in a common framework, Sparck Jones [1999] introduces a model of so-called "context factors". Context factors can be used to tailor a summarization system to a specific area of application as they describe the desired target summary for this application and thus help choosing a development strategy for the summarization system.

Three kinds of top-level context factors are distinguished, each of which is further divided into sub-factors as shown in Figure 2.1. The idea behind this schematic is that it describes the *function* of a summary: "given Input Factor data, to satisfy Purpose Factor requirements, via Output Factor devices". It is important to note, that the key element in this framework is to provide structural guidance for strategic decisions to be made in the development of a summarization system, and not so much to provide a complete "check list" of factors that will determine a single best choice for a summarization system. For instance, the set of factors shown in Figure 2.1 are not concerned with the distinction between extractive and abstractive methods which could be an example for a further factor to add to the schema.

It is interesting to see how our own task of automatic meeting summaries can be analyzed according to context factors.

Input Factors

Form The input to a meeting summarizer naturally are meetings. The *structure* of meetings vary, from fairly loose (e.g., brainstorming sessions) to strictly formalized (e.g., parliamentary debates). A predetermined agenda is a typical way for people to structure their meetings, but even in cases where an agenda is used, there are differences in how much room the participants allow for divergence from the agenda.

The *scale* factor describes the size of the content to summarize and how it affects the way a summary has to be conceived. Summarizing a monograph

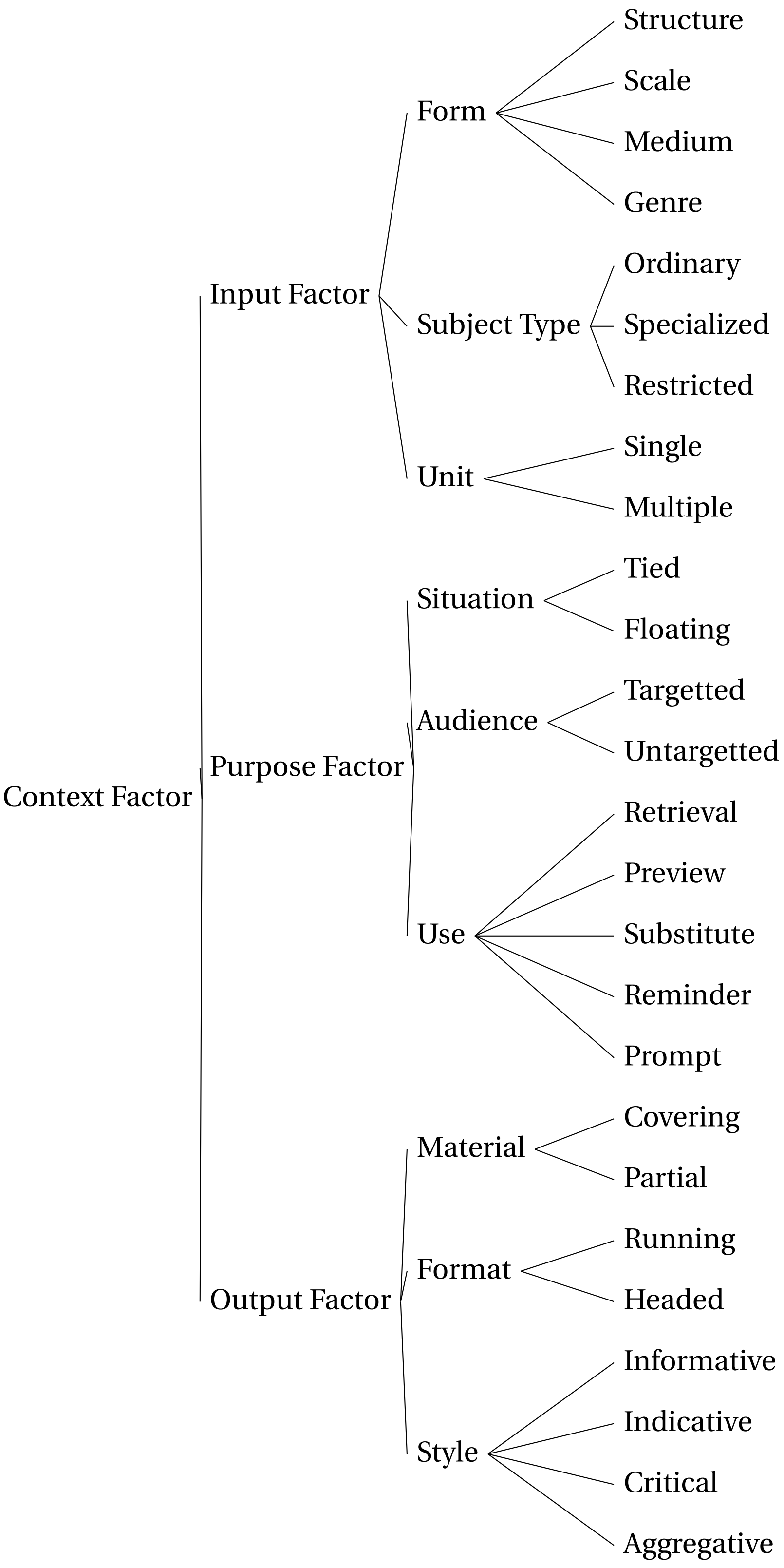


Figure 2.1: Context factors according to Sparck Jones [1999]

may call raise different constraints on compression as well as the extent of content transformation than summarizing a single paragraph.

The original intention for Sparck Jones *medium* factor was that of describing the language and potential sub-language(s) used in the source. However, we can extend this notion to *modalities* and distinguish between categories such as text, audio, audio/video.

Genre is a factor that describes different literary forms, such as, narratives, technical reports, news stories. It is clear that the difference between these forms can have a wide-ranging influence on the way how to summarize a document.

Subject Type The distinction Sparck Jones gives for the two cases *specialized* and *restricted* is surprising, as it depends on the physical location of the consumers. In both cases, the subject matter of a source is not expected to be generally known, but limited to a smaller group of people. If this limitation is due to a regional constraint (e.g., a local newspaper), it falls in the restricted case, otherwise it is called specialized (e.g., an article from an international chemistry journal). If a source is neither specialized nor restricted, it is called *ordinary.*

Unit The *unit* factor refers to the distinction between single and multi-source summaries, as described above.

Purpose Factors

Situation The situational factor can be seen as an abstraction of the question addressing query-specific and generic summaries which we discussed before. However, it also includes shades in between those two extremes, for instance, a summary which although not created to answer a specific query is crafted with a specific target use in mind. As an example, we a summary of a new product description made for the company's marketing department. We would call such a summary *tied* to a particular situation. The more general case without a particular context in mind, is said to address a *floating* situation.

Audience A similar notion is that of the target audience. This factor, however, refers to the group of people that a summary is expected to reach. A book review on the World Wide Web can be said to be relatively *untargetted* because it may in principle be consumed by any person. In contrast, an abstract of a scientific article can be expected to be consumed mainly by a specific clientele. Therefore we call such an abstract *targetted.*

Use This factor addresses the question what a summary will eventually be used for. Among the possible applications of a summary, we find *retrieving* a particular source document, *previewing* a document to get a quick overview, *sub-*

stituting the source document altogether, a *refreshing* reminder of an already known source, or as action *prompts* that invite to consume the summary's source.

The *use* factor is related to the *style* output factor. The difference is a matter of perspective: *use* describes the intended usage at the time of creation of the summary while *style* may vary in assessment from one consumer to the next one.

Output Factors

Material Depending on whether a summary concentrates only on certain aspects of the source or tries to capture all of the important aspects equally. In the first case, a summary is called *partial*, in the second case *covering*. An example for a partial summary in the context of meeting summarization would by a summary that only reports on the decisions made during a meeting.

Format The *format* factor addresses the structure of a summary. While often times, summaries come in the form of *running* text, it is conceivable that they divide their contents up under different headings (hence *headed* summary). For instance, the gold-standard summaries manually written by annotators of the AMI corpus follow a specific slot structure (cf. chapter 4.4). The abstracts typically found at the beginning of scientific papers are, by contrast, typically running summaries.

Style The output factor *style* covers the distinctions between *informative*, *indicative*, *critical*, and *aggregative* as discussed in sections 2.6 and 2.9.

Table 2.1 summarizes the actual values for the context factors as used in the concrete approach of this thesis. At this point, this shall only serve as an overview or point of reference as the concrete motivation for the values will be given in the following chapters.

Table 2.1: Classification of presented approach according to context factors

Input Factors	Form	Structure: *Agenda/Free discussion* Scale: *30min discussion* Medium: *Multi-modal* Genre: *Multi-party interaction*
	Subject Type	*Ordinary*
	Unit	*Single*
Purpose Factors	Situation	*Floating*
	Audience	*Untargetted*
	Use	*Retrieval/Substitute/Refreshing*
Output Factors	Material	*Covering*
	Format	*Running*
	Style	*Informative/Indicative*

Chapter 3

Related Work

3.1 Introduction

In this chapter, we give an overview of the relevant research carried out to date. Both manual and automatic summarization have been studied in the past. Most of the previous work has concentrated on document summarization and so literature dealing with automatic meeting summarization is quite scarce. Furthermore, researchers often favor extractive approaches for their apparent ease of methodology. As a result, this thesis is, to the best of my knowledge, the first complete approach to abstractive meeting summarization. Abstractive approaches have successfully been applied to other domains tough–again, mostly within document summarization– and we may draw general insights from them that could prove useful for the meeting domain as well.

We begin with a short survey of how people use summarization in their everyday lives to get a good understanding of the involved mental processes that may serve as a blueprint or guideline for our own automatic approach (Section 3.2).

After that we report on interesting work on automatic document summarization in Section 3.3. Although not all of the results achieved there can be re-used for meetings because of the fundamental differences in the underlying domains (documents vs. meetings), it still deserves careful inspection as it is by far the most extensively studied area of automatic summarization. For the same reason, we highlight some outstanding results from extractive summarization work, despite the fundamentally different nature of such approaches.

As with extractive summarization, the earlier works on abstractive summarization were carried out on documents. Section 3.4 gives an overview over selected approaches.

Although hardly any abstractive approaches exist for meeting summarization, there are a few interesting results available for extractive summarization of meetings. Naturally, such approaches imply the use of meeting transcripts (manual or automatic) from which they extract text material. The most relevant results will be given in Section 3.5.

Section 3.6 addresses previous work on meeting summarization. We find that previous work in that area is rather limited, mostly because of a tight dependence on a particular application domain.

To wrap up the chapter, we critically discuss all of these findings and how they relate to our own task at hand in Section 3.7. We conclude what can be learned or even taken over from past research to abstractive meeting summarization.

3.2 Human Summarization

That summarization is indeed a difficult task becomes clear when we look at the cognitive processes that are involved when people summarize texts or events they have experienced. Endres-Niggemeyer [1998] gives an excellent overview of the psychological theories that help explain human summarization.

In human-human communication, summarization occurs both explicitly and implicitly. Not only does the human mind constantly abstract information when partaking in an ongoing discourse, summaries are also often used instruments of everyday interaction, e.g., telling a friend about the latest movie, reporting recent results in a staff meeting, etc. The most prominent difference is that one is an unconscious process (abstracting during discourse understanding) while the other one is an active task (creating summaries for a consumer). The following sections explain both types of human summarization in detail.

Discourse Understanding

When we observe a situation or participate in a communication, we build a mental representation of what we perceive ("constructive assumption", [van Dijk and Kintsch, 1983]). We are, however, only able to retain a very limited amount of information in our short term memory at a time during discourse interpretation [Miller, 1956]. Consequently, without an ability to mentally abstract information on the fly, we would not be able to follow a normal conversation or understand a text we read[1] in real-time.

Knowledge Processing Strategies

In order to cope with the memory limitations during discourse understanding, people have to have ways to reduce the representation of the communicated contents in their minds. According to van Dijk and Kintsch, we apply various knowledge processing strategies concurrently to do so. The term "strategy" may sound as if referring to a conscious activity, such as solving a puzzle etc., but it is really meant in this context as a cognitive procedure that occurs in the brain without (the requirement of) active control. It does, however, hint at a notion that these procedures are

[1] For reasons of simplicity, we take the view of reading or listening to a text as a specialized form of communication here, in which a producer (the author) communicates in a one-way direction with the consumer (the reader) via a medium (the text). We will not go into further details of the broad field of *Communication Theory*.

goal-driven, i.e., a strategy describes “an efficient way to reach a specific goal”–for instance, abstracting information.

Among the strategies outlined by Endres-Niggemeyer [1998] are *propositional strategies* which transfer statements in the input source into a propositional representation. Propositions formulate properties of entities and how different entities relate to each other. For example, a sentence in a newspaper article such as “Rahn scored a goal” might be represented in the reader’s mind by the conjunction of three propositions: `Event(Goal)` $\wedge$ `Person(Rahn)` $\wedge$ `scorer(Goal, Rahn)`[2]. To arrive at such a representation when reading this sample sentence is the result of applying a propositional strategy.

Strategies of knowledge use and *inference strategies* work in an intertwined fashion. For instance, in the above example, we might retrieve the fact `Person(Rahn)` from memory if we are familiar with soccer players. In that case, we would make use of prior knowledge. However, even if we didn’t know this particular player, we would still be able to infer the same proposition because we know that only players can score goals and that all players are people. Thus, more general knowledge together with inference rules can lead us to the same understanding.

If in the same article we read “It was in the 85th minute” right after the above example sentence, *strategies of local coherence* together with *schema strategies* would allow us to understand that Rahn’s goal in that game was scored five minutes before the end. This is especially notable as neither the word “game” nor any similar concept is mentioned. For such an inference, we would use knowledge about how newspaper articles of sports events are typically written and that goals are often reported along with the time at which they occurred. Such knowledge about typical courses of events for a given situation has been coined *schema* by Schank and Abelson [1977].

The most important strategies from the perspective of summarization are *macro-strategies*: they use and combine certain inference rules called *macro-rules* to compile multiple propositions into more abstract representations. These resulting representations are called *macro-propositions*. The process in which macro-rules are replied to yield macro-proposition can be applied recursively to yield higher degrees of abstraction. The final structure is called a *macro-structure* [van Dijk, 1980]. Van Dijk and Kintsch [1983, p.199] identify three such macro-rules:

- *Deletion*: Given a sequence of propositions, delete each proposition that is not an interpretation condition (e.g., a presupposition) for another proposition in the sequence.

  We can also look at this rule from a more positive point of view. Instead of leaving certain propositions out, we can consider keeping certain propositions, in which case we call the reverse macro-rule *Selection*.

[2] Note that the tense of the original sentence was not modeled here for reasons of simplicity.

- *Abstraction*: Given a sequence of propositions, substitute the sequence by a proposition that is entailed by each of the propositions of the sequence.

  In earlier works, this rule is sometimes called *Generalization*.

- *Construction*: Given a sequence of propositions, replace it by a proposition that is entailed by the joint set of propositions of the sequence.

Explicit Summarization

In strict terms, the above theories of human summarization during discourse understanding differ from what would colloquially be referred to as "summarizing". Most notably, no *summary document* is produced during the process, everything happens exclusively within the human mind. However, the key task performed here is that of *content reduction* which naturally corresponds with summarization, and so the study of such theories are relevant if our goal is to create a system that is able to produce near human-quality summaries automatically.

For such an automatic system, it is particularly interesting to study the different reduction strategies outlined by Kintsch et al. On an abstract level, they inspire processing steps that could be part of a computer-based system. Especially the ideas of a *propositional representation* and of transformation operations on top of such a representation is appealing because it is closely related to what computers are designed to do: store and manipulate information.

Unfortunately, the psychological theories only claim the existence of such strategies but they do not explain in sufficient detail *how* the strategies achieve their respective goals. Further studies are required in order to help closing this gap–and also to gain insights not only into general abstracting methods, but also how to eventually create a summary document.

Endres-Niggemeyer [1998] describes the process of summarization as a three step activity (s. Figure 3.1). The starting point of the summarization process is the source which is to be summarized. The summarizer consumes the source (reading, hearing, watching, etc.) and by doing so *understands* the contents of the source. The result of this understanding step is a representation of the meaning of the source, internal to the summarizer. This representation is then *reduced* to yield a representation of the summary contents. In a last step this internal summary representation is again externalized by creating some sort of summary *presentation*, e.g., a written abstract.

One should note that this depiction is a high-level abstraction of the summarization process–as we have shown in the previous sections, the different representations are not necessarily as clearly separated as depicted in the figure and likewise the three steps are not performed clearly sequentially but occur partly in parallel. The questions that this model raises are: how do the three transition phases (understanding, reduction, presentation) progress and how are the internal states of

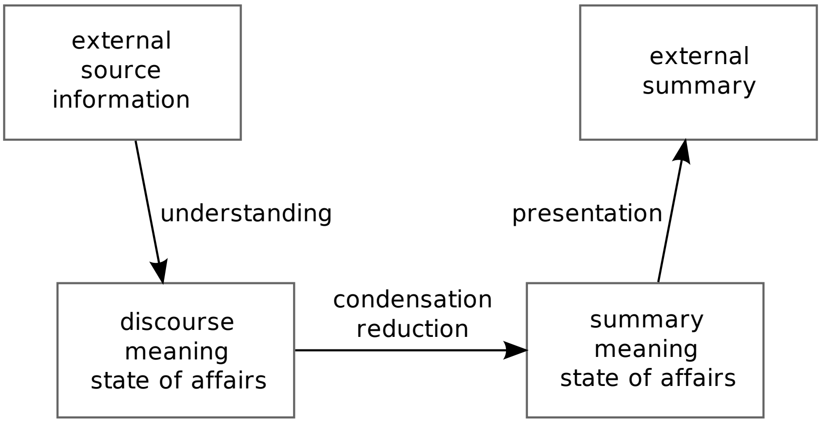


Figure 3.1: The summarization process: sub-tasks and intermediate result states. [Endres-Niggemeyer, 1998]

affairs represented. For the latter question, Endres-Niggemeyer lists and explains a number of well-known units of representation (basic units, such as, categories, concepts, relations, propositions, and larger units, such as schemata, frames, scripts, and memory organization packets) and how these different ways of representing are integrated.

Variations of this model exist. For instance, the three steps are alternatively called *interpretation*, *transformation*, *generation* in [Sparck Jones, 1999]. [Cremmins, 1996] names four stages *interpretation*, *selection*, *reinterpretation*, *synthesis*. In this thesis we will use Spärck Jones' terminology.

Discussion

The insights of Kintsch and Endres-Niggemeyer are in so far interesting as they may serve as guidelines in the development of a suitable representation formalism for an automated summarization system. Like a human understander, a meeting summarization system is confronted with the task of representing the contents of discourse in order to transform them into a representation of summary contents. One of the key challenges in automatic abstracting thus is to find a suitable representation formalism (cf. chapter 5.4).

Human summarization requires a substantive amount of world-, domain-, and procedural knowledge. While research in Artificial Intelligence (AI) has produced impressive results and continues to do so in many important yet specialized subtasks, to simulate the exact processes of the human mind–even only in so far as they are understood by today's standards–is a challenge beyond the state of the art in the field.

A logical follow-up question is whether the human approach to summarization is the only feasible one, or if other methods exist that lead to comparable or at least acceptable results while being more suitable for machine implementations. For instance, as we see below, extractive summarization (see 3.3) relaxes our idea of form and coherence of a summary; similarly, automatic abstractive approaches (see 3.4) typically constrain the domain of the source domain to which they can be applied. In conclusion, if we allow slight modifications of our understanding of the summarization task, it becomes more tractable for computational approaches. This will be illustrated in the following sections.

3.3 Extractive Document Summarization

It is fair to say that in the history of document summarization, two papers stand out when considering their impact on subsequent research. Thus, the seminal works by Luhn [1958] and Edmundson [1969] receive a rather extensive treatment in this section. To this day, their ideas influence ongoing research in extractive summarization and many if not most of later research is based on their original ideas. To give a comprehensive review of over fifty years of research would go beyond the scope of this section, but we will highlight some of the influential papers which contributed important ideas to the field.

In 1958, H.P. Luhn established the field of automatic summarization by developing the extraction approach to document summarization [Luhn, 1958]. Limited by the computational capabilities of its time, his work already features a number of important ideas and concepts that are the basis for the development for an entire new field of research. In his method, the sentences of a document are assigned a significance factor which is a function of the words it is made off. For that, a list of all words of a document is compiled and sorted by frequency after stemming the words with a simple heuristic. Stop words are removed based on their frequency of occurrence: words with very high frequency are considered too common to contribute significance to the sentences they appear in and words with very low frequency are taken to be too irrelevant.[3] Therefore, an upper and a lower bound are defined and only the words within this interval are considered significant from there on. The significant words in a sentence are clustered together if they are separated by no more than four insignificant words. Then, a significance value is computed for these clusters by dividing the square of the significant words in a cluster by the total number of words in the cluster. A sentence's significant value is then defined as the maximum value of all its contained clusters. For the final summary, all sentences that rank higher then a certain threshold are extracted.

[3] For the removal of stop words, Luhn discusses both the possibility of using a pre-compiled list and detecting such words solely based on their high frequency in the document.

The key observation made by Luhn is that the frequency of words in a document can be taken as a measure for their relevance. This idea proved to be fruitful for other applications as well, perhaps most notably in information retrieval (e.g. "tf.idf", [Salton and Buckley, 1988]). The limited computational capabilities in his day, however, may have been the reason why Luhn took a few pragmatic shortcuts. For instance, he decided to treat the significance of a word as a Boolean factor: a word is either significant or insignificant, thus discarding the differences in the apparent Zipf-like distribution of words. Since the significance factor of a word is ultimately the basis for all significance computations, this seems to somewhat contradict his own argument of a correlation between significance of a term and its frequency–there is no notion of some words being more relevant than others. Likewise, how he derives clusters from the significant words of a sentence could be criticized as being too crude, as it ignores phrase boundaries and can produce quite unintuitive results. Consider, for instance, a sentence with the following pattern:

```
+ - - - - + +
```

where a plus stands for a significant and a minus for an insignificant words. Such a pattern would define a single cluster, yielding a sentence value of $3^2/7 = 1.29$ according to Luhn's formula. However, by inserting another *insignificant* word in the middle of the sentence, we would actually *increase* the value of the sentence. This is because with more than four insignificant words between them the beginning and the end of the sentence would fall into two separate clusters, one of value 1 and one of value 2, with the latter one giving the overall sentence value. This result is partly due to the somewhat ad-hoc formula used to compute a cluster's significance value.

A decade later, Edmundson [1969] conducted an extensive study to address the ad-hoc aspects of Luhn's work and juxtapose them with insights taken from observing human summarization results. To gain these observations, Edmundson bases his work on two different corpora of scientific articles, one–called the "heterogeneous corpus"–for collecting *a priori* statistical data (common words, sentence lengths, etc.) and one for the actual extraction experiments. For the creation of a *gold standard* of article summaries, the human summarizers receives detailed instructions by which to extract sentences from the source documents. These instructions contain not only content-related rules, but also pay attention to aspects such as redundancy and coherence.

For the automatic extraction, the task is to determine characteristics of the document that correlate positively with extracted sentences in the gold standard summaries. Four different types of features (called "methods") are considered by Edmundson:

Cue words Four lists are compiled from the heterogeneous corpus: *null* (words that appear in a large number of all documents in the corpus and are therefore considered insignificant), *bonus* (words with a high relative frequency in gold-standard sentences), *stigma* (words with a low relative frequency in gold-standard sentences), and *residue* (words that although not insignificant do not clearly tend to correlate either positively or negatively with gold-standard sentences). The cue weight C of a sentence is the sum of the cue weights of the words it contains.

Term frequency Residue-words with a frequency within the document above a certain threshold are considered more relevant than those with low frequency. The key weight K of a sentence is the sum of the frequency of all such words contained in it.

Title term Non-null words that appear in headings in the document receive a positive weight, and a sentence containing such words receives the sum of their weights as its final title weight T.

Location This feature consists of two parts. In one part, it assigns weights to sentences depending on their position in the document and in their position in the paragraph they appear in. A second part uses a pre-compiled dictionary of common section titles ("Introduction", "Conclusion", etc.) and assigns weights to sentences that appear under such headings. The exact weights are determined statistically using the corpus. The sum of the weights from both parts of this feature determines the sentence's overall location weight L.

To arrive at one single score $W(s)$ for a sentence s, Edmundson combines the four features as a weighted sum: $W(s) = a_1C(s) + a_2K(s) + a_3T(s) + a_4L(s)$. He performed various experiments to derive good combinations of the weights a_i and found that his system performed best in a combination of C,T and L, leaving out the method based on term frequency.

Edmundson's work was groundbreaking because it introduced a number of novelties. He observed that both structure and content of a document can provide relevant hints for the extraction of single sentences. In addition, he realized that although some features are document-specific, some characteristics can be drawn from a corpus of similar documents. He viewed his own work as an instance of the four possible combinations of these two dimensions which he called "structural" and "linguistic":

		Structural Features	
		Document Body	Skeleton of Document
Linguistic Features	General Corpus Characteristics	C	L
	Specific Document Characteristics	K	T

Interestingly, the best combination of features he found left out the term frequency method, despite the common intuition that a term's relevance correlates with its frequency in the source. It appears that the other three types of features, when combined, can outweigh the relevance information present in a term's frequency alone.

Edmundson's work set up a framework for subsequent research: the idea of representing a sentence as a combination of separate features allows to view later work as two interrelated tasks: finding good features on the one hand and good methods to combine them on the other hand.

A lot of different approaches have followed the tradition of Luhn and Edmundson (cf. Mani and Maybury [1999] for an overview), where the main focus is laid on two research question: how to produce a sentence ranking that best reflects the sentences' relevance and how to combine the highest scoring sentences from this list into the final extractive summary. Today, especially for the first task, the use of machine learning and statistical classification has become a popular approach. In such systems, sentences are considered data points that are classified according to how summary-worthy they are. This might be a strictly Boolean decision or a real-valued confidence number. The general method for such approaches is to use an annotated corpus, i.e., a collection of documents together with manually crafted extractive summaries, and derive a prediction model from these data. The exact details of this vary, but in any case the resulting model can be used for a system which takes a document as input, classifies the sentences of the document according to the model and creates an automatic extract from the classification result.

Among the first to advocate such an approach were Kupiec et al. [1995]. They estimate the probability of each sentence in the source to be included in a summary based on five types of sentence-level features F_i: *minimum sentence length*, *contained cue phrases*, *position in paragraph*, *contained high frequency words*, and *contained proper names*. The summary-worthiness of a sentence s is then estimated as a conditional probability $P(s \in S|F_1,...,F_k)$. Using Bayes' rule this can be computed from values taken from the training set if statistical independence between features is assumed[4].

As a variation of Kupiec's Bayesian classifier, Teufel and Moens [1999] use a handcrafted list of indicator phrases that mark types of meta-comments in scientific articles, such as, "we argued" or "in this article". The presence of such a phrase in a sentence becomes a feature in addition to length, location, *tf.idf*-like, title words and paragraph type features. Also, each indicator phrase is manually assigned one of 16 rhetorical classes (see section 3.3), however, this *indicator rhetorics feature* was left out in the final system because it was found to slightly decrease the overall system performance.

For the evaluation of their system, the authors measure the recall of the gold-standard summary sentences that their system would extract. The addition of the indicator phrase list proves to be a valuable extension of Kupiec et al.'s original feature set: while the best individual result for the other features is a recall of 39.6% (*header type feature*), the indicator phrase feature reaches a score of 54.4%. The best feature combination yields a value of 66.0%.

Myaeng and Jang [1999] employ a similar idea, but instead of combining features prior to the classification, they classify all features separately with Bayes classifiers and combine the results using the Dempster-Shafer combination rule [Shafer, 1990]. This allows to rank sentences according to their estimated probability and create the output summary from the highest ranked sentences.

As features they use the thematic role a sentence plays with respect to the entire document (*background*, *main theme*, *explanation of document structure*, *future work*). Since the vast majority of the selected gold-standard sentences belong fill the role *main theme*, they can alternatively filter out those sentences that do not belong to that role. Other features for sentences are *contained high frequency words*, *positive and negative words*, *position in document*, *resemblance with title*, and *comparison of sentence vector with document vector*. For the last feature, both the sentence and the whole document are represented in a vector-space model and this feature is used to estimate how central the sentence is to the document's subject matter.

The same representation is also used for another interesting aspect, the elimination of redundant sentences. Here, the rationale is to avoid repetitions which make the summary look unnatural and also use the available summary space suboptimally. If the vector similarity of two candidate sentences is above a certain threshold, the one with the lower rank score is omitted from the summary. This thought poses an important development, as it steps out of the line given by approaches that classify each sentence "out of context" of what has already been found to be summary-worthy.

[4] The authors do not discuss whether this is a realistic assumption–it is arguable whether feature pairs such as *minimum sentence length* and *contained cue phrases* or *contained cue phrases* and *position in paragraph* are in fact independent.

The use of term-based features, such as cue phrases or frequency counting, is limited by the fact that authors of documents typically vary the used vocabulary for reasons of style and to avoid redundancy. For instance, they may use nominalizations and verbal forms of a word interchangeably ("decision", "decide), or make use of synonyms ("result", "outcome") and hypernyms ("remote control", "device"). Classification features that rely on a pure surface level of matching fail to detect the similarity between such variations and thus are likely to miss important relatedness information. In particular, if the frequency counts are not accumulated in some way, the different variations of word forms will each receive a lower count.

One way to cope with such issues is the use of a thesaurus, such as, WordNet [Fellbaum, 1998], that encodes relations like the ones above between lexical units. In principal, such a term categorization enables a document parser to map different terms to the same canonical representation [Hovy and Lin, 1999]. Instead of using the original terms, features based on cue phrases or frequency counting fall back on these canonical terms. For example, if a text contains the term "decision" once and "decide" also once, both would be mapped to a canonical term which then would get a frequency count of 2.

Depending on the subject matter of a document, it is clear that certain terms naturally appear more often than others if they are central to the document's topic. Not always, however, does that necessarily mean that they are especially information baring and therefore good markers for extract-worthiness. For instance, as we discussed above, cue phrases such as "in this paper" can be very good predictors in scientific articles. If we imagine, however, a report from a paper factory about a newly developed material, the term "paper" has a different meaning and thus loses much of its discriminative power with respect to sentence salience. In addition to thesaurus-based techniques, one way to deal with this problem is to accompany pure term frequency within the source document with a measure of how significant a term is to the general subject matter. A standard way to do this in the field of Information Retrieval is to use a corpus of (more or less) similar documents and compute for each term the *inverse document frequency* [Jones, 1972; Van Rijsbergen, 1979], a measure for how seldom a term appears in the corpus. This techniques has also been taken over to automatic summarization [Aone et al., 1999; Hovy and Lin, 1999].

But the use of a training corpus also allows to compute other useful information. Aone et al. [1999] use an external newspaper corpus to compute statistics of co-occurrences of two subsequent nouns. The idea is to automatically derive statistical knowledge about compounds, such as, "potato chip", in order to be able to distinguish from expressions that are similar on the surface ("computer chip") but very different in meaning. The key observation here is that single words may not be the ideal atomic unit upon which to operate, since multi-word expressions can not simply be decomposed into their constituents. Another technique used by the authors that is aimed in the same direction is to employ a *named entity tagger* to find occurrences of fixed name expression (people, places, companies, etc.).

Relaxing the idea of two-word co-occurrence further, we can compute more general *association patterns* between words, i.e., sets of words that tend to often appear together in the same document. Hovy and Lin [1999] take such patterns to represent a concept of "topic". They, too, make use of WordNet to grasp concept relatedness beyond the level of a concrete lexical unit. For each occurrence of a term in the document, the term's weight is increased by one as is the weight of every term that appears above this term in the WordNet hierarchy. This technique boosts the weight of more general concepts in the hierarchy. Therefore a second step traverses the hierarchy in the opposite direction–from top to bottom–to find the most specific concepts for the description of the document's content. At each node, beginning with the root, if one of the child nodes has a weight clearly higher than the weights of its siblings, this child is selected. Otherwise, the process is continued recursively for all child nodes. As a result, one obtains a horizontal "cut" through the concept hierarchy, i.e., a list of selected nodes that are good abstractions of their descendant nodes, yet not overly general. This list is called the *interesting wavefront* [Lin, 1997].

Discourse Structure

One general difficulty extractive summarization approaches face is the fact that the sentences in a source document are embedded in the context of discourse. When taken out of this context, extracted sentence may thus lack information that is required in order to understand their content. For instance, extracted sentence may contain anaphoric references to entities from other sentences. If the references information is not also present in the final extract, understanding of the summary contents may be quite difficult to a reader who is not aware of the source document.

The discourse structure of a document can be viewed on different levels of granularity. On a "microscopic" level, such structure is called *cohesion* [Halliday and Hasan, 1976] and can be achieved through a number of different devices, such as, anaphors, ellipses, or conjunctions.

Barzilay and Elhadad [1997] try to account for text cohesion by using lexical chains for text summarization. In their approach, they analyze the source text to identify nouns, compounds and adjective–noun collocations. For these lexical entities, they look up all the different senses available in WordNet. For each new sense thus discovered, their system tries to link it to the senses found in previous parts of the text using certain relations, such as *synonym*, *antonym*, etc. This procedure generates "chains" of linked WordNet senses. The algorithm of Barzilay and Elhadad then uses these chains to identify important topic threads within the document. Using heuristics such as *extract a sentence if it is the first to mention a chain item*, these lexical chains are used to identify extract-worthy sentences for the final summary. These heuristics can be combined with a ranking scheme for chains based on features such as chain length or the relative number of distinct occurrences. The ranking can be used to control the length of the produced summary by incorporating only those sentences that stem from the highest scoring chains.

Besides such approaches based on text “cohesion” one can also study the discourse structure of a document on a more “macroscopic” level, which is then usually referred to as "coherence".

Marcu [1997] analyzes the discourse structure of documents on the basis of Mann and Thompson’s Rhetorical Structure Theory (RST) [Mann and Thompson, 1988] which segments a text into non overlapping spans with certain relations holding between them. A segment is either a *nucleus* or a *satellite* where the nucleus expresses essential information complemented by the information in the satellite. The rhetorical relations between nucleus and satellite are, for example, *Contrast*, *Elaboration*, etc. These relations are such that the information inherent in the nucleus can be understood without that inherent in the satellite while the opposite direction is not necessarily true. Recursive application of the relational analysis yields nested spans which induce a discourse tree for the underlying text.

For summarization, it is speculated by Marcu that the satellites of a discourse tree can be omitted. Thus, the remaining nuclei form a reduced version of the original text, the summary. Marcu developed an algorithm that can derive an RST tree of a document automatically. The resulting tree nodes are ranked according to the rhetorical relations encoded in the tree and the top ranked nodes are extracted until the desired summary length is reached.

Marcu developed his approach on written documents. However, as [Stent, 2000] discusses, the application of RST to dialogs is not without caveats which makes the applicability of Marcu’s approach to meeting summarization questionable. In particular, Stent criticizes the ambiguity of some RST relations when applied in a concrete annotation effort. At the same time, she observes that discourse effects such as adjacency pairs (see Section) could not be easily expressed with RST relations.

A different approach is taken by Strzalkowski et al. [1998]. In an analogy to information retrieval methods, they automatically construct a document “query” from terms identified using common features, such as *tf.idf* score, positioning in the document, cue phrases etc. The document is then segmented into paragraph-like passages, and passages are scored according to term-overlap with the query, normalized by the length of a passage. The top ranked combination of passages that still satisfies a maximum length criterion for the final summary are certain to be extracted.

Passages, however, may contain referential “backlinks” (sic) to previous passages and if a passage with a backlink is extracted, the linked passage is extracted as well. This constraint naturally has a great influence on the overall extraction result.

The motivation for these backlinks is straightforward: to foster text coherence, the authors want to assert that background information from the source document is included whenever necessary for the general understanding of the discourse. Therefore a central point of Strzalokowski et al.'s work is the development of methods to automatically detect reference to background passages and insert backlinks to the referred to passages accordingly. The methods to decide if a passage provides background or novel information include:

- finding anaphors such as pronouns and definite noun phrases
- detecting partial names
- identifying rhetorical relations through explicit grammatical markers, such as, conjunctions and adverbials, but potentially also RST relations.

Once identified, these properties can be treated in different ways. For instance, given a passage that was found to contain a backward reference, one heuristic to find the correct background passage is to simply use the immediately preceding passage. But instead of inserting backlinks, an alternative is to simply remove the referring expressions, for example by replacing a pronoun with the full name of the person referred to, or by removing cohesion markers such as trailing conjunctions. Such an approach is sometimes called "text rewriting" [Nenkova, 2008].

Teufel and Moens [1999] introduce a similar idea, but extend the division into *background* and *what-is-new* into a more fine-grained template. Their approach is an extension of Kupiec et al.'s Bayesian classifier described above. On top of the automatic extraction results produced by that classifier, a second classification step is performed in which each extracted sentences is categorized as to what *rhetorical role* it plays in the source document. Here, a rhetorical role is one of

- BACKGROUND
- TOPIC/ABOUTNESS
- RELATED WORK
- PURPOSE/PROBLEM
- SOLUTION/METHOD
- RESULT
- CONCLUSION/CLAIM

It is clear that such a classification presents a form of structuring a document. For automatic summaries, despite being shallow, it allows for a more flexible and variable summarization method that could better tailor its output to a particular user and what aspects of a document exactly he is interested in, e.g. through controlling the amount of background information to be included etc.

Unfortunately, Teufel et al. never actually implemented the final summarization system on the basis of their Bayesian classifier, perhaps because the overall classification results of their approach leave room for improvement: the classifiers correctly extracts and classifies 42.3% of the gold-standard sentences.

3.4 Abstractive Document Summarization

When we compare the techniques described so far, which concentrate on finding the most salient sentences from a document, with the steps performed by a human summarizer to arrive at a summary, we find that the two processes do not have a lot in common. With respect to Spärck Jones' general view of summarizing, we observe that the *transformation* and *generation* phases are certainly underrepresented in extractive approaches: generation is in most cases merely a verbatim copying of material originally conceived by the author of the source document, while transformation typically only consists of computing a sentence score and selecting the top-ranked sentences. This may partly be owed to the fact that the *interpretation* phase in extractive summarization consists of little more than mapping the source sentence to feature vectors. Such a sparse representation leaves little room for complex transformation and generation actions. However, some researches have used cognitive models like the ones outlined above (section 3.2) as an inspiration for summarization system with more profound transformation and generation phases.

The FRUMP system [DeJong, 1982] utilizes two ideas by Schank, *scripts* [Schank and Abelson, 1977] and *conceptual dependency* [Schank, 1972, 1975], to represent the contents of newswire articles. Based on this representation, the system can generate cross-lingual summaries of the source documents. The idea of scripts is extended to the notion of *sketchy scripts* which differ from Schank's original *script* concept (cf. Section 4.3). in that sketchy scripts omit less important events and only represent what is conceived as the most important sub-events of the described event. An example for a sketchy script describing the event of a police arrest is shown in figure 3.2.

FRUMP's sketchy scripts are implemented as conceptual dependency structures. Consequently, the FRUMP system faces the limitations inherent in this representation formalism, such as the lack of quantifiable assertions. Sketchy scripts can be understood as relatively simple templates with slots to be filled by a natural language understanding component which in case of FRUMP consists of two major sub-components, called the *predictor* and the *substantiator*, which are based on [DeJong, 1979].

The basic idea of the *predictor/substantiator* approach is that text parsing is guided by a semantic representation of the current context of the text at hand. Based on this context, predictions are generated about what kind of information the system expects to encounter next in the source. These predictions are generated by

```
$ARREST:
1. Police go to where the subject is.
2. There is an optional fighting between the suspect
   and police.
3. The suspect is apprehended.
4. The suspect is taken to the police station.
5. The suspect is charged.
6. The suspect is incarcerated or released on bond.
```

Figure 3.2: An example of a sketchy script as used by the FRUMP system.

the *predictor* component. In turn, it is the *substantiator*'s task to find evidences for the current predictions. There are two basic kinds of possible evidences: direct evidence in the text and evidences inferred logically from existing knowledge.

Direct evidence for a sketchy script means that the system finds surface constructions directly in the source text that are sufficient for the activation of this script ("*Explicit Reference activation*"). For instance, if the text contains the noun phrase "the crime", FRUMP will activate the *crime* sketchy script.

In case no evidence can be extracted directly from the source text, FRUMP has two ways to infer evidences from previously extracted information. For one, sketchy scripts are not represented internally as isolated from each other. Rather, FRUMP stores causal relations between different scripts and whenever a script is activated which the system knows to often precede a certain other script, the second one is activated, too ("*Implicit Reference activation*"). For instance, if a newspaper report activates a sketchy script representing *crime*, the *predictor* will also activate a script for *arrest* because the system expects the report to possibly contain information about an arrest as well. The second indirect activation method is based on the idea that some sub-events might be so central to a sketchy script that finding evidence for such a sub-event is sufficient to also activate the sketchy script itself ("*Event-Induced activation*").

If the *substantiator* is able to find an evidence for a predicted fact, either directly or indirectly, the model of the current context is updated accordingly and predictions are refined or generated anew. If no evidence can be found, the *predictor* re-assesses the previous predictions and adapts them before the *substantiator* attempts again to find evidences for these new predictions.

One special case of the *substantiator* fulfilling a prediction is its way to handle anaphors: if the evidence found in the text for a predicted role filler is an anaphoric expression, such as, a pronoun, FRUMP will take this as an indication that the hypothesized role filler was predicted correctly, as long as their is no contradiction regarding number and gender between the pronoun and the predicted role filler.

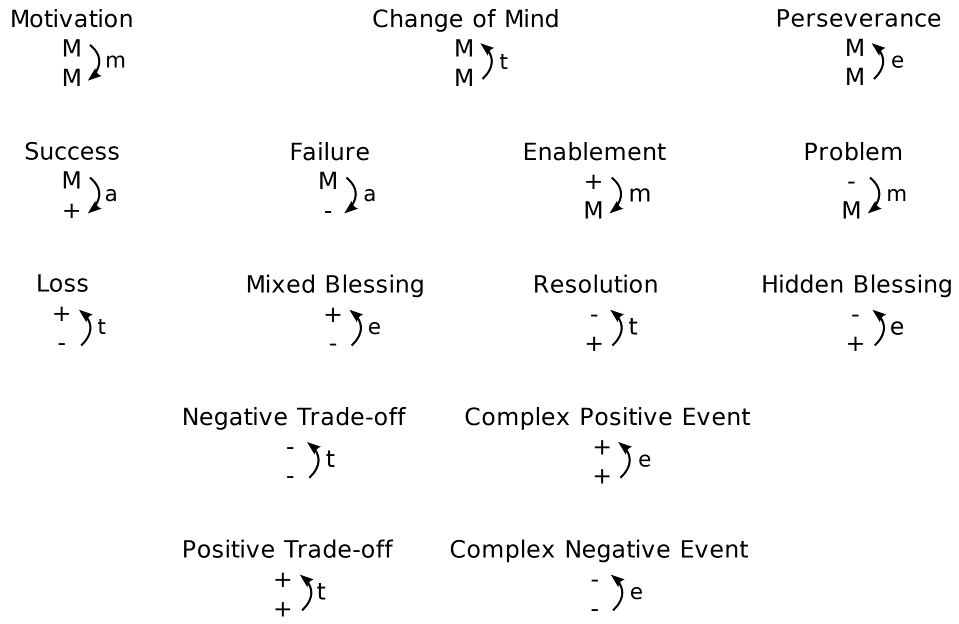


Figure 3.3: Primitive plot units

Unlike FRUMP, Lehnert [1981] approaches content interpretation in a bottom-up fashion. She analyzes narratives to yield a chronological order of two types of atomic *affect states* which represent either *mental states* (M) of one of the protagonists or positive (+) and negative (-) *events* that happen in the narrative, including actions carried out by the protagonists. To model the relations between these states, four types of causal links are introduced, Motivation (m, describing causalities, must point to a mental state), Actualization (a, describing intentionality, must point from a mental state to an event)), Termination (t, ending of an affect state, can only link states of the same type), and Equivalence (e, multiple perspectives of an effect state, can only link states of the same type).

By enumerating all possible combinations of two affect states and the possible links between them, Lehnert arrives at 15 primitive plot units (cf. figure 3.3). Assuming a chronological order from top to bottom, the primitive plot unit Success, for instance, describes an event that was intended by the protagonist at some previous point in time and which came out positive for the protagonist. Another example would be the Loss unit in which a negative event ends a prior positive event.

These primitive units build the basis for more complex patterns consisting of more than two states and thus combining multiple primitive units into larger plot units. Two examples are shown here, Intentional Problem Resolution (problem & success & resolution) and Starting Over (success & loss & problem & perseverance).

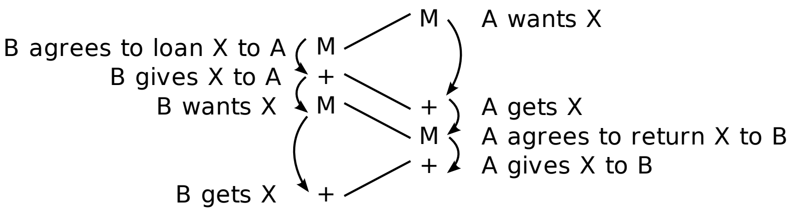
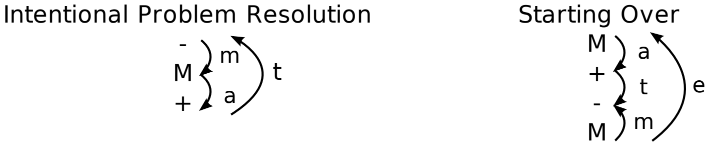


Figure 3.4: Plot unit analysis of a situation involving two characters *A* and *B*



These patterns are character-specific, i.e., each character in the narrative receives its own analysis of affective states and causal links between them. For stories with multiple characters that means that we will have several analyses in parallel. To mark interactions and interdependencies, so-called *crosscharacter* (sic) causal links are introduced. Figure 3.4 depicts the analysis for a situation in which a character *B* loans something *X* to character *A*. The left column describes *B*'s perspective on the situation and the right column *A*'s perspective. We have two types of crosscharacter links in this example, `M` — `M` (REQUEST) and + — + (SHARED EVENTS) and the combination of a REQUEST and a SHARED EVENTS in the reverse direction is taken by Lehnert to be a complex unit HONORED REQUEST. Therefore, the example contains two HONORED REQUESTS, the combination of which is called EXCHANGE

In similar ways it is possible to construct patterns of growing size that denote increasingly complex interactions one may find in narratives. These patterns form an abstraction of the low-level events in a narrative that can be used as the basis for summarizing the narrative.

Although limited to narratives, Lehnert's model is the first to address summarizing the interaction between multiple protagonists. On the way to meeting summarization, Alexandersson [2003] moves one step further with his approach to automatically generate summaries of negotiation dialogs [Reithinger et al., 2000]. The studied domain is taken from the VERBMOBIL project [Wahlster, 2000], an automatic translation system of dialogs between two people making an appointment. The two speakers have a short dialog in which they agree on a time and place to meet as well as activities to engage in.

Alexandersson makes use of the available interpretation infrastructure that is part of the VERBMOBIL translator and adds on top of this a module for summary generation. The representation in his SUGE module consists of an *intention* part and an *content* part. For the former, each contribution of a speaker is broken down into dialog act segments (cf. 4.4), and categorized automatically into one of 19 different types of dialog acts [Reithinger and Klesen, 1997]. These dialog acts model intentional aspects of a speaker contribution, such as *Inform*, *Request*, etc.

The actual contents of these contributions are realized on a second level. The propositions of a dialog act segment are represented in so-called discourse representation expressions (DIREX). These expressions contain expressions of different kinds. The modeling of the necessary domain knowledge for making appointments is realized using *description logic*. In addition, temporal aspects are encoded with temporal expressions using the specialized TEL language [Endriss, 1998]. SUGE also employs a notion of a general topic of discourse which for the given domain is one of four possibilities: *Scheduling* (finding a date for the appointment), *Traveling* (arranging modes of transportation), *Accommodation* (where to stay), *Entertainment* (which spare time activities to engage in).

Altogether, an utterance is therefore analyzed in terms of three aspects, as illustrated by the following sample sentence:

"So we have to leave Munich at six o'clock"

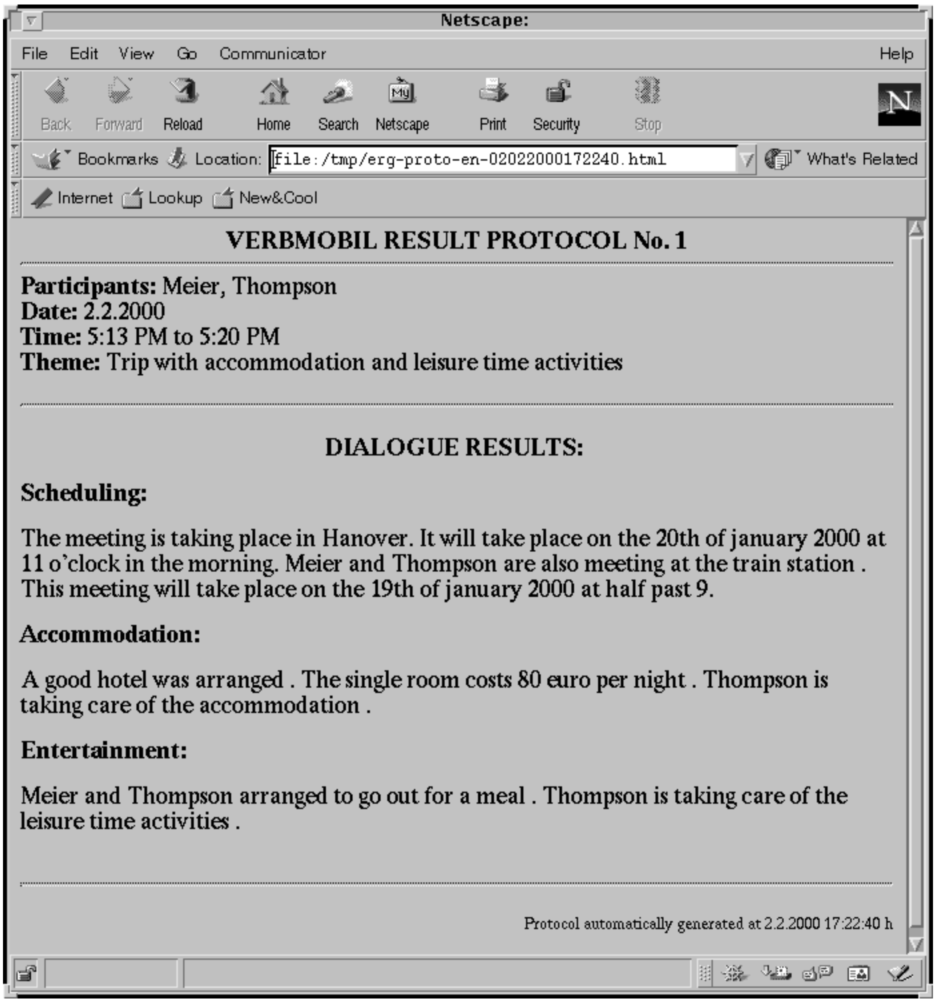
Dialog act: *Suggest*
Topic: *Scheduling*
DIREX:

```
has move:
[move, has_source_location:
       [city, has_name='muenchen'],
       has_departure_time:
       [date, tempex='tempex(en_2920_0,
                            [from:tod:6:0])']]
```

With this kind of representation, Alexandersson first analyzes all speaker contributions of a negotiation dialog. His system is able to follow the negotiation sub-dialogs in the different topics and can accumulate the resulting negotiation objects by combining the *Direx* with a default unification algorithm [Alexandersson et al., 2006]. Then, a summary is generated in four steps. First, the most recent specific accepted negotiation objects are compiled from the dialog analysis. For each, a plan processor generates a sentence representations for each negotiation object. A smoothing step introduces anaphors and demonstratives to obtain a more pleasant and cohesive text style. In a final step, this sentence level representation is passed to VERBMOBIL's generator VM-GECO which realizes the sentence structures as text. Figure 3.5 displays an example summary as generated by Alexandersson's summarizer.

3.5 Extractive Meeting Summarization

One of the biggest problems for adapting an approach such as Edmundson's and similar ones to (extractive) meeting summarization is that they rely to a certain de-


Netscape:
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Stop
Bookmarks
What's Related
Internet Lookup New&Cool

Location: file:/tmp/erg-proto-en-02022000172240.html

VERBMOBIL RESULT PROTOCOL No. 1

Participants: Meier, Thompson
Date: 2.2.2000
Time: 5:13 PM to 5:20 PM
Theme: Trip with accommodation and leisure time activities

DIALOGUE RESULTS:

Scheduling:

The meeting is taking place in Hanover. It will take place on the 20th of january 2000 at 11 o'clock in the morning. Meier and Thompson are also meeting at the train station . This meeting will take place on the 19th of january 2000 at half past 9.

Accommodation:

A good hotel was arranged . The single room costs 80 euro per night . Thompson is taking care of the accommodation .

Entertainment:

Meier and Thompson arranged to go out for a meal . Thompson is taking care of the leisure time activities .

Protocol automatically generated at 2.2.2000 17:22:40 h

Figure 3.5: Example of a Verbmobil dialog summary [Wahlster, 2000, p516].

gree on the availability of structural information such as paragraphs, sections and section titles, etc. That this observation is especially true for approaches that explicitly employ discourse roles for analyzing text documents goes without saying. A transcript of a meeting of course does not come with such nice structuring hints.

Another problem is that in a fully-automatic system, linguistic features are subject to the quality of the automatic speech recognition (ASR) system used to create the automatic transcript. If certain cue terms that could add to a sentence's relevance score are said during a meeting but misrecognized by the ASR system, we expect the extraction quality of an automatic summarizer to degrade. Even worse is the effect ASR deficiencies have on the final extract: the text quality is directly influenced by the text quality of the automatic transcript.

Nevertheless, some research has been carried out on extractive meeting summarization in recent years. Disregarding considerations like the above, Waibel et al. [1998] use a modified version of a maximal marginal relevance (MMR) approach originally developed for text summarization on meeting transcripts. They construct an extractive summary by iteratively adding the highest ranked speaker turn until a desired length is reached. Their approach merely applies standard document summarization techniques to a meeting transcript.

Zechner [2001] pays more respect to the fundamental differences between a carefully drafted document and a transcript of spontaneous dialog. He introduces a seven stage pipeline architecture for his DIASUMM system which extracts a summary from a transcript of a dialog. The first five stages in this setup address the question of determining useful units in the transcript which then serve as the basis for a subsequent extraction procedure. A part-of-speech (POS) tagger first assigns to each word in the transcript with a tag from an extended version of the Penn Treebank POS tag set [Marcus et al., 1994]. The extension adds four special markers for *disfluent* words [cf. Shriberg, 1994]. These tags are used in the next stage to determine reasonable "sentence" boundaries. This is a challenging problem in that people in spoken discourse do not always produce full and/or grammatically correct sentences. Therefore, the third stage tries to detect another class of speech disfluencies, false starts. Similar to the discourse-oriented document summarization approaches described above, Zechner notes that some of the transcript units available for extraction can not be understood in isolation, but should be presented together with other units. One example for such cross-speaker information links are question/answer-pairs, and consequently DIASUMM's fourth stage detects such pairs. The next module in the pipeline removes another type of speech disfluencies within a speaker's utterance, repetition, of lengths between one and four words. A longer dialog may range over multiple topics and it might be desirable for a summary not to leave some of them out. For an automatic system, it is therefore useful to detect topics boundaries within a dialog transcript. This is what the fifth stage in Zechner's system does, the last stage before the actual extraction of sentences. Extraction is realized in two steps, ranking the sentences and selecting the most relevant ones. Each sentence is represented as a term vector and likewise a term vector is computed for each topic segment. Based on this setup, all sentences in a topic segment can be ranked according to their vector's similarity to the segment vector. To account for the aforementioned cross-speaker information links, sentences that are members of question/answer pairs are always ranked at adjacent positions in the list, at the position of the higher-ranking pair member. With the sentence ranking in place, DIASUMM supports three variations of selection procedures with differences in how the sentences or parts of them are presented to the use.

Murray [2008] elaborates on the last step of Zechner's pipeline, the actual extraction step. While Zechner uses three variations of a *tf.idf* as the basis for his computations, Murray investigates whether such term-weighting metrics that were originally intended for documents can and should be used for spontaneous speech summarization as well, or whether using more specialized metrics tailored to better capture the differences in language could produce improved results. He introduces two new metrics for that, *su.idf* and *twssd* and compares them to four standard metrics from Information Retrieval (*idf*, *tf.idf*, *ridf*, and *Gain*).

The first term weighting scheme, *su.idf*, is based on the intuition that the different speakers that contribute to the meeting contents each has a personal vocabulary at their disposal that differs from that of the other speakers. His hypothesis is that the speakers will use more informative words with a higher variation in frequency while words that occur with a more evenly balanced frequency between the speakers will be considered less informative. He therefore formulates a *surprisal score* for each term t and each speaker s, as the negative log probability that the term is uttered by the other speakers:

$$surp(s,t) = -\log\left(\frac{\sum_{s'\neq s} tf(t,s')}{\sum_{s'\neq s} N(s')}\right)$$

A total surprisal score $totsurp$ for each term can then be computed by summing over all speakers-dependent surprisal scores. The final formula multiplies the result with the relative number of speakers who actually utter a given term ($s(t)/S$) and the square-root of the inverse document frequency idf:

$$su.idf(t) = totsurp(t) \cdot \frac{s(t)}{S} \cdot \sqrt{idf(t)}$$

The second metric *twssd*[5] is a variation of the same intuition, namely that informative words are not evenly distributed amongst all speakers. In turn, this has a consequence on the probability that a randomly chosen term from the meeting transcript was uttered by a specific speaker. Using Bayes' Theorem one can estimate this probabilities on the grounds of readily available statistics from a training corpus. Using for each term t the maximum of the scores thus estimated for all speakers in a meeting yields a score Sc_1 that will be higher for more "personalized" terms and lower for commonly used terms.

To account for structural properties in a meeting discourse, a second score Sc_2 is introduced, in which Murray measures how evenly a term is distributed over all speaker turns in the meeting. This is also in analogy to Sc_1, but substituting speaker information with turn information.

In a third component score Sc_3 Murray computes co-occurrence statistics of each term from the used corpus, as he hypothesized that this might be a valuable information for term-weighting. With these three component scores *twssd* is computed as the harmonic mean of Sc_1, Sc_2, and Sc_3. For purely term-based extraction, Murray finds that both of his newly introduced metrics outperform traditional schemes from document summarization when evaluating the generated

[5] *Term-weighting for spontaneously spoken dialogues*

summaries with the gold-standard summaries on the basis of recall, precision and f-score of co-selected sentences.

As previous research on extractive summarization has shown cue words to be valuable hints for extract-worthy sentences, Murray encompasses his term-weighting approach with a list of manually derived cue words. An initial list is ranked based on how well their correlation with extracted and their inverse correlation with non-extracted gold-standard sentences. Experiments based on this simple approach alone already showed f-scores comparable to the experiments based purely on term-weighing, although precision was observed to drop.

Murray then studies whether results could be improved by combining different kinds of features. He proceeds to compare three different unsupervised machine learning approaches for this, MMR, Latent Semantic Analysis (LSA, [Deerwester et al., 1990]), and a variation thereof based on the Centroid approach by Radev et al. [2000]. Although MMR and LSA receive a notable gain when using Murray's *su.idf* term-weighting scheme compared to using traditional *tf.idf*, the Centroid method outperforms both, showing no significant difference between *su.idf* and *tf.idf*.

Also, supervised machine learning approaches using a liblinear logistic regression classifier are studied. In addition to term-weighting and cue word features, he also includes prosodic features derived from the audio recordings of the speakers (mean and maximum energy; mean, maximum and standard deviation of F0 pitch; dialog act lengths; pause lengths, dialog act overlap lengths; and an estimation of rate-of-speech), structural features (position of dialog act in the meeting; position of dialog act within a speaker turn), and features estimating speaker dominance. He finds that a subset of features from all classes delivers best results.

3.6 Abstractive Meeting Summarization

Although extractive meeting summarization has seen growing interest in the research community in the last decade, abstractive approaches are still scarce. Traditionally, extractive approaches have always outnumbered the research on abstractive summarization, in document as well as speech-based summarization. We believe this to be due to the relatively clear and well-defined task of extraction that lends itself well to machine learning and statistical classification, versus the more complicated and complex task of abstraction which has in the past relied on error-prone NLP methods. With the source material being less clean and structured in the case of meetings, it is understandable that the trend toward extractive summarization has only continued. A second reason may be that abstractive summarization requires a domain representation formalism. Often times, this implies dependence of a certain domain of discourse. The implication of low re-usability in other domains may be another reason why abstractive summarization is considered less attractive.

Notable exceptions are rare. Castronovo [2009] describes a knowledge-based approach for summarizing design decisions in meetings. He models the domain of discourse (the design of a remote control with different design aspects, such as, its color, material, type of batteries, etc.) explicitly in an ontology. The interpretation of the discussions in a meeting is an incremental process, interpretation results are successively propagated through three interpretation layers: In the *extraction layer* every speaker contribution is parsed with a semantic grammar parser [Engel, 2006], resulting in a Typed Feature Structure that represents the current utterance. If applicable, this structure is merged on an *intermediate layer* with previous utterances on the same topic. Lastly, the *result layer* collects the final state of all topics from the intermediate layer and integrates them into a complete representation of the current remote control design.

Similar to other knowledge-based approaches, Castronovo's work suffers mostly from its strong dependency on a hand-crafted domain model. While such a model enables him to produce summaries of good quality, it makes it difficult to apply his approach to other domains without adjusting the used ontology, typed feature structures, semantic parser rules, etc. The labour intensity as well as the expert knowledge required to perform these adjustments make the approach unfeasible for wider application.

An alternative approach that attempts to deplete the dependence on a specific domain of discourse while still aiming for an abstractive summarization model is presented by [Murray et al., 2010]. Like Castronovo, they manually design an ontology, but only as a model of conversations, not of a particular domain. It consists of only two main categories, *Participant* and *Entity*, which can be specialized by subcategories such as e.g. *ProjectManager*. The focus of this ontology lies on a set of six relations between instances of the two main categories. Relations are realized as triples of the form:

<*Participant, Relation, Entity*>

expressing that a particular participant in the conversation stands in the given relation to an entity that is mentioned in the discourse. For example, the following triple expresses that a certain participant is involved in a decision about a simple chip: <*participant-a, hasDecision, simple-chip*>. Such triples are instantiated by several classifiers that take as input a sentence from the conversation and output triples according to the six relations which model decisions, actions, problems, positive subjective sentences, negative subjective sentences and generally important sentences that do not belong to any of the other relations. Entities are detected as noun phrases with a medium-range (10-90%) frequency, and participants are known through speaker identification.

Murray and colleagues argue that this kind of representation is suitable for downstream processing by an NLG component that could generate an abstract. However,

> *The meeting was opened and the meeting group talked about the user interface, the remote control and the design. They debated the costs, the company and the project while discussing the project budget. The signal, the remote control and the beep were mentioned afterwards. They talked about meeting before closing the meeting.*

Figure 3.6: An indicative abstractive meeting summary generated by the system of Kleinbauer et al. [2007a].

they only demonstrate how the triple representation can be leveraged for extraction. Once all triples have been instantiated in an *interpretation* step, their *transformation* step uses an integer linear programming method to optimize an extraction function.

This approach is hybrid to some extent, because it combines a symbolic representation with a sentence extraction method. The representation formalism has two major advantages. It is not domain-dependent, modeling only conversations but not the topics of discourse, and it is simple enough to be feasible for the interpretation phase, as the evaluation done by the authors suggests. However, since they have not actually implemented the *generation* phase, it is unclear how eligible their relatively shallow ontological representation is for such a task. For instance it is questionable how well their ontology lends itself to macro-rules such as van Dijk and Kintsch [1983]'s *Abstraction* and *Construction* (cf. Section 3.2).

Our own previous work is so far the only attempt at realizing a complete abstractive summarization pipeline that includes a text generation phase [Kleinbauer et al., 2007a,b]. It makes use of a elaborate domain ontology [Lochert et al., 2005] to encode the propositional content of meeting transcripts. A second layer of information is a hierarchical set of discourse topics, e.g., *discussion, presentation of prototypes, project budget, look and usability, etc.*. The topic information gives a basic structuring of the discourse, and each topic segment is realized as a distinct sentence that verbalizes the general topic of the segment as well as the main points of discourse.

For the latter, we use the most frequent concepts that appear in the ontological propositional content annotations of a topic segments. A presentation planner which has access to that information creates the general structure of the summary document. The result of the planning phase are logical forms that serve as input to the surface realizer NIPS [Engel, 2006]. The result of the generation phase is exemplified in Figure 3.6.

The ontology used in this work is orders of magnitude larger than the one in [Murray et al., 2010]. But it also demonstrates the previously discussed discrepancy between expressiveness of a representation scheme and the difficulty to populate such a scheme with instances from the discourse: the propositional content and topic annotations used by the summarizer are created manually. The *interpretation* phase is thus not fully automatic, even though the downstream phases are. This makes the summarizer semi-automatic and thus not feasible for practical use.

3.7 Discussion

In this chapter we have presented an overview of the state of the art of automatic summary generation, spanning from the historic roots in the mid-20th century to today. We have looked at research addressing different aspects as well as different genres of source documents. Here, we argued that speech, and more so meeting summarization is still a relatively young field that has not yet received as much attention as classical document summarization. Throughout the literature, extractive approaches dominate the field, and although some research has attempted generating abstractive summaries, those works remained spotlight approaches, failing to attract wide-spread attention in the research community. In particular, abstractive summarization of meetings has not been tackled at all.

In addition to the work presented above, more exotic takes on summarization existSuch work includes, for example, summarization of singled-out aspects of meetings (decisions [Hsueh, 2008; Wang and Cardie, 2011] or action items [Purver et al., 2007]). An interesting alternative to text-only summaries of meetings is presented by Castronovo et al. [2008] who develop a constraint-based layout engine to visualize meeting summaries in either a newspaper or a comic-strip style (Fig. 3.7). Sometimes the generation of textual descriptions of numeric event data, e.g. [Maybury, 1995], is also referred to as summarization. However, we argue that this task is only remotely related to meeting summarization, if at all.

For our own work, we have looked at findings from psychology and cognitive science that could provide useful insights for establishing a novel approach to abstractive meeting summarization. In particular, we have derived a general abstract model for the summarization process that will serve as the basis for the remain work in this thesis. Endres-Niggemeyer's work described both mental representations of information and cognitive strategies that play a role in human summarization.

On a more technical side, we have highlighted the central concerns to machine-based summarization. The main question of extractive summarization is how to determine which sentences to extract from the source. For meeting summarization, this question has to be reformulated slightly because spontaneous speech is in most cases not made of full (grammatical) sentences.

In any case, once a unit or segment for extraction is found, the next step is to establish good indicators for the salience of a segment. For documents, features such as term frequency, cue words, and position of a sentence have been shown

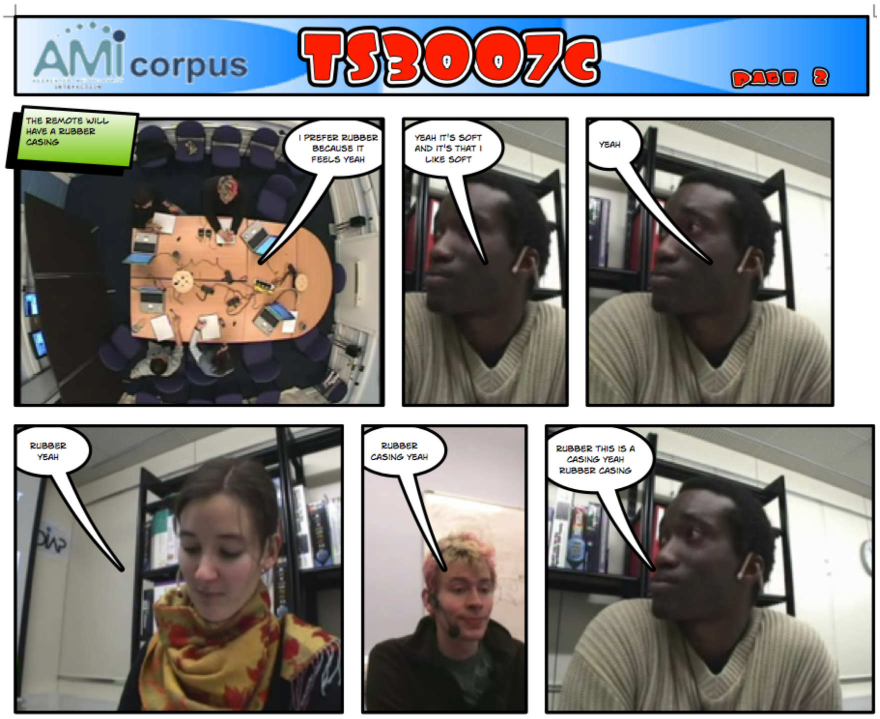

AMI corpus
TS3007c
PAGE 2
THE REMOTE WILL HAVE A RUBBER CASING
I PREFER RUBBER BECAUSE IT FEELS YEAH
YEAH IT'S SOFT AND IT'S THAT I LIKE SOFT
YEAH
RUBBER YEAH
RUBBER CASING YEAH
RUBBER THIS IS A CASING YEAH RUBBER CASING

Figure 3.7: An excerpt of a comic-strip summarization generated by the SuVi system [Castronovo et al., 2008].

to be the most useful. For multiparty conversations, Murray shows that variations of established term weights prove beneficial. Combining features to the reach best extraction results is nowadays almost always approached with machine learning approaches.

The output of such computations are ranked lists of original sentences, and while earlier approaches contented themselves with extracting the top scoring sentences off the list, much work has been spent since then into incorporating discourse information to increase the stand-alone readability of the produces summaries.

Abstractive approaches are still comparatively rare. Explicit modeling of a domain of application and the inherent restriction to that domain may be a reason for this negligence. We have described two genuinely different approaches above, a top-down approach using script-like representation structures and a bottom-up approach which uses three simple affect states to model complex structures emerging from interactions of protagonists in narratives. For speech, Alexandersson introduces a method to generate abstracts of negotiation dialogs.

Since abstractive summary approaches rely on a meaning analysis of the source document, they are prone to a type of error that can not be found in extractive summaries: if the source is misinterpreted somehow, the final abstract will contain wrong information, i.e., describe facts that did not actually occur in the source (e.g., "confabulations" [Alexandersson, 2003]). Yet, extractive summaries, if not carefully constructed, may display a similar kind of error: two sentences that stem from different parts of the documents and are taken out of context through the extraction process, could together form a new meaning never expressed in the source. This happens, for example, when an extracted sentence contains an anaphora which in the summary could be interpreted to refer to something different than in the source document. Figure 3.8 exemplifies the effect in a summary generated by [Murray, 2008]' extractive summarizer: the "it" at the beginning of the second line suggests that the tool training exercise needs to be original, trendy and user-friendly. In the original meeting though, these attributes refer to the remote control which the meeting group is designing.

PM: then we'll move into acquaintance such as getting to know each other a little bit, including a tool training exercise.

PM: it needs to be original, trendy and user-friendly.

Figure 3.8: Extractive confabulations: dangling anaphors suggest contents that were never actually uttered in the source meeting.

One possible way to deal with the shortcomings of either of the two approaches is to try to combine the best of both worlds, and some researchers have gone this way. Coming from the extractive end of the scale, Hovy and Lin [1999] for example make use of the WordNet hierarchy to map pure surface forms of lexical units to *synset* representations (sets of synonymous words) which allows them to exploit inter-conceptual relationships such as hypernym/hyponym. This can be interpreted at a first small step towards a richer representation of contents which supports very simple reasoning facilities. Our own previous work [Kleinbauer et al., 2007a,b] can be viewed as a similar approach in which frequency statistics that are typical for extraction features are used in a semi-automatic abstractive meeting summarizer.

Table 3.1 highlights and compares a selection of the different approaches discussed above. The two classic approaches by Luhn and Edmundson paved the way for research on automatic summarization, and can certainly be said to have established the field. However, both of their approaches relied heavily on manual tweaking. DeJong's FRUMP system was the first serious attempt to generate summaries in a more human-inspired fashion, by using a symbolic representation for the summary contents. Unfortunately, the summaries were presented only as internal datastructures of the system, and not in textual form. Besides that, FRUMP was limited to a single sketchy strip per input which makes the approach unsuitable for meetings which typically encompass multiple different topics.

Kupiec realized that extractive summarization could be treated as a binary classification task. Today, most research on summarization follows in the footsteps of this early machine learning approach; however, like all extractive approaches, cohesion and coherence of the produced textual output is a challenge, and even more so, if his approach were translated to a less-structured and more freely developing domain such as spontaneous conversations.

Hovy et al. attempt to use a more conceptually rich representation for their summarizer than simple feature vectors. In particular, they suggest the use of already existing language resources, such as WordNet, to achieve both a broad conceptual coverage and a more symbolic representation. Their system is constructed to support multiple different text generation components, however, such components are not presented by the authors.

Zechner suggests an extraction approach for meetings and other types of conversations. He addresses the spontaneous nature of the source by correcting some speech disfluencies automatically. However, in the end he does not generate a new summary text, but merely extracts speaker contributions from the transcript.

In contrast, Alexandersson's approach does use a text generation component. In particular, his system is able to leverage the translation capabilities of the VERBMOBIL system to generate cross-lingual summaries, albeit in only a quite limited domain.

Our own previous work faced similar challenges in terms of coverage. Although the textual quality was good due to its abstractive nature, the analysis of a given meeting was partly done by hand because automatic components failed to delivered the required quality.

Murray's approach is similar to Zechner's but improves upon it in a number of points by employing a model more catered to speech-specific features. It remains an extractive summarizer though that does not improve the inherent difficulties of extraction over transcripts.

Finally, the work presented in this thesis makes a novel contribution by being a complete, automatic system for the abstractive summarization of group meetings. The chosen representation formalism, Frame Semantics, is rich enough to allow for content reduction operations similar to how humans summarize. At the same time, the formalism is not too ambitious for real-life use, as is underlined by the existence of different software parsers for Frame Semantics. Yet, the sometimes lacking coverage of FRAMENET and of the Scripts used as part of the content reduction phase still limit the overall approach.

Table 3.1: Comparison of selected summarization systems.

	Extractive or Abstractive	Application	Content Reduction	Generation	Novelty	Drawbacks
Luhn (1958)	E	Technical papers and magazine articles	Ranking threshold	no generation	First automatic summarization	Shallow representation, ad-hoc heuristic
Edmundson (1969)	E	Scientific articles	Weighted feature sum	no generation	Multiple heterogeneous features, gold-standard evaluation	Hand-written feature combination
DeJong (1982)	A	Newswire stories	Script activation	no generation	Script representation	Only one sketchy script per story
Kupiec et al. (1995)	E	Technical articles	Probability estimation	no generation	Extraction as binary classification	Ignores cohesion and coherence
Marcu (1997)	E	Newspaper articles	Satellite omission	no generation	RST	Suitability for discourse unclear
Hovy et al. (1999)	E/A	Text documents	Word clustering	not implemented	Concept fusion, “wavefront”	Low precision and recall
Zechner (2001)	E	Telephone dialogs, TV news, group meetings	Sentence similarity to topic vector	no generation	Speech-based summarization	Cohesion / coherence not addressed
Alexandersson (2003)	A	Face-to-face and telephone dialogs	“Complete” operation	SuGe+GECO	Abstractive, multilingual conversations	Limited coverage
Kleinbauer et al.(2007)	A	Group meetings	Concept frequency	Template-based generator	First meeting abstractor	Semi-automatic
Murray (2008)	E	Group meetings	Maximum marginal relevance	no generation	SU.IDF, extraction over ASR	No treatment of cohesion / coherence
Kleinbauer (2011–*this thesis*)	A	Group meetings	Macro-rules	Partial syntax trees	Use of Frame Semantics	Limited by FrameNet coverage

Chapter 4

Knowledge Sources and Data Sets

4.1 Introduction

The approach to abstractive meeting summarization described in this thesis suggest the integration of some well-founded theories. The goal of this chapter is to introduce the relevant formalisms.

Section 5.4 introduces Frame Semantics, a theory of lexical semantics, modeling the meaning of words. To summarize a meeting, we concentrate mainly on what the meeting participant said to each other.

One source of information that we can leverage in the transformation from source to summary contents is *script* knowledge which describes the sub-events that make up more complex events. This theory is described in Section 4.3.

We use existing data sets to develop and test our approach as well as extracting certain knowledge bases, such as e.g. the partial syntax trees used for text generation (Section 5.6). These data sets–the FRAMENET corpus, the AMI and the AMIDA corpus–are introduced in Sections 4.4 and 4.4 respectively.

Finally, we summarize these preliminaries in Section 4.5.

4.2 Semantic Frames

In a meeting, people communicate not only through words, but also through other modalities, such as hand and body gestures, facial expressions, presentations, etc. The meeting participants make use of the different modalities in an intertwined way, i.e., a single utterance may transport information that is distributes over multiple modalities. For instance, consider the following excerpt of meeting ES2002a from the AMI corpus:

In this part of the meeting, participant D draws a beagle on the whiteboard in the meeting room. This action is not very clear from the transcript, even though speaker B introduces the general topic in the first three lines of the excerpt. D volunteers to be the first in this experiment, but we see no indication in the transcript that he actually gets up from the table and walks over to the whiteboard to draw a beagle. We only learn that a beagle is his favorite animal. Therefore, in order to summarize this part of the meeting correctly, a summarizer would have to monitor

1 **B:** Um and at this point we get try out the whiteboard over there .
2 **B:** Um .
3 **B:** So uh you get to draw your favourite animal and sum up your favourite characteristics of it .
4 **B:** So who would like to go first ?
5 **D:** I will go . That's fine .
6 **B:** Very good .
7 **D:** Alright . So
8 **D:** .
9 **D:** This one here , right ?
10 **B:** Mm-hmm .
11 **D:** Okay . Very nice .
12 **D:** Alright .
13 **D:** My favourite animal is like
14 **D:** .
15 **D:** .
16 **D:** A beagle .

Figure 4.1: Excerpt of a manual transcript of meeting ES2002a from the AMI Corpus.

visual channels too, in addition to only the verbal exchange.

In line 9, D asks a clarification question: *This one here , right ?*. Again, from the transcript alone, it is not clear what he refers to, but the video reveals that there are two whiteboards in the meeting room. D reassures that he is using the correct one, by combining a pointing gesture with a verbal question.

These examples show that not only can some information be available solely in non-verbal modalities, sometimes a correct interpretation of the discourse requires a multimodal analysis. However, we argue that the relative infrequency of non-verbal actions in comparison to verbal exchanges justifies a concentration on the transcript as the main source for automatic abstracting.

Our main concern for the interpretation phase of summarization will thus be of linguistic nature, i.e. how to arrive at a content representation from the spoken discourse.

Lexical semantics is the study of the meaning of words. Since single words are sometimes not sufficient to transport meaning (e.g., *the*, *by*), while sometimes the same word may carry multiple meaning options:

1a. The window is *open.* (state)
1b. Could you please *open* the window? (action)
2a. The *bank* is closed on Saturdays. (institution)
2b. The party takes place in a former *bank.* (building)

In sentences 1a. and 1b. the word *open* refers to a state and an action respectively. We see that they differ in their part of speech (POS)[1]: the *open* in sentence 1a. is an adjective, the one in sentence 1b. a verb. This distinction, however, is not sufficient to differentiate between word meanings, as is exemplified in sentences 2a. and 2b. Here the word *bank* is a noun in both cases, yet 2a. refers to the institution while sentence 2b. refers to a building.

We thus retract to a more precise term as the basis for our analyses, called *lexical unit*. Cruse [1986] defines lexical units as

> *[...] the smallest parts which satisfy the following two criteria:*
>
> 1. *a lexical unit must be at least one semantic constituent*
> 2. *a lexical unit must be at least one word*

We typically denote a lexical unit in the form *lemma.pos*, i.e. the lemma, a dot, and the part of speech in lower-case letters. For instance, the lexical units from the above examples are denoted as *open.a*, *open.v*, and *bank.n*.

Linguistic semantics is a topic of ongoing research, a number of semantic theories exist in parallel and are actively worked on by different researchers. One of these theories, *Frame Semantics*, was first introduced by Fillmore [1976]. While classic semantic theories are inspired by formal logic and attempt to represent meaning through truth conditions, Fillmore argues that in order to understand the meaning of a lexical unit, one must understand the contextual situation that unit refers to. In Frame Semantics, such situations are represented by the *Frame*, a concept inspired by Minsky's notion of frames [Minsky, 1975], Schank's scripts [Schank and Abelson, 1977] and Fillmore's own previous work on case grammar [Fillmore, 1968].

Frame Semantics' central stance is that linguistic knowledge used in language production and understanding is not isolated from other sorts of knowledge. Instead, people are assumed to have means for storing and accessing a collection of cognitive schemata, or Frames, for structuring, classifying and interpreting experiences.

[1] We use either the POS tags of the PENN Treebank Project (see Appendix B), or for smaller examples, the following simplified list: **A**, **N**, **V**, for **a**djectives, **n**ouns, and **v**erbs.

A (semantic) Frame is a "schematic representation of a situation involving various participants, props, and other conceptual roles, each of which is a frame element" [Fillmore and Petruck, 2003]. When a communicator hears or reads a certain word, a Frame, i.e. an abstract situational reference, is evoked in the speakers mind. Other words can then be understood by assigning the referenced entities roles from the evoked Frame. Take for instance the following sentence (cf. [Petruck, 1997]):

Carla bought the computer from Sally for $100.

Here, the word *bought* evokes the COMMERCE_BUY Frame which is defined in Figure 4.2[2].

Here, a short prose definition is given for the Frame itself as well as for the different Frame Elements (FE's), i.e. semantic roles, that this Frame contains. The lexical unit which evokes a Frame is called the *Target*. The other parts of the sentence can be interpreted as Frame Elements of the COMMERCE_BUY frame (note that by convention we include in our annotation the full phrase a FE appears in):

Buyer:	Carla
Goods:	the computer
Seller:	from Sally
Money	for $100.

Thus the evoked situation in this sentence is that where someone (Carla) buys something (the computer) from somebody (Sally) for a specific price ($100). Note that we don't know who Carla and Sally are and which computer is referred to. Frame Semantics makes no statement about the identity of the referenced entities, but provides structures through which these entities are related to each other.

A Frame can thus be used to describe the meaning of some linguistic material when we assume that we have the means to disambiguate the referenced entities. That makes Frame Semantics a good method to represent the contents of texts or conversations.

Further annotated example sentences from the FRAMENET corpus (see Section 4.4) follow, where we use a bold face to mark the target and the colors from Figure 4.2 to mark Frame Elements.

1. She gleefully **bought** the rock.
2. Will they allow you to **purchase** by check?
3. I have been **buying** from him for over 10 years.

[2] `http://framenet.icsi.berkeley.edu/fnReports/data/frameIndex.xml?frame=Commerce_buy`

Frame

Commerce_buy

Definition

These are words describing a basic commercial transaction involving a buyer and a seller exchanging money and goods, taking the perspective of the buyer. The words vary individually in the patterns of frame element realization they allow. For example, the typical pattern for the verb *buy.v* is: Buyer buys Goods from Seller for Money.

Frame Elements

Buyer:	The Buyer wants the Goods and offers Money to a Seller in exchange for them.
Goods:	The FE Goods is anything (including labor or time, for example) which is exchanged for Money in a transaction.
Manner *(Manner)*:	Any description of the purchasing event which is not covered by more specific FEs, including secondary effects (quietly, loudly), and general descriptions comparing events (the same way). It may also indicate salient characteristics of the Buyer that affect the action (presumptuously, coldly, deliberately, eagerly, carefully).
Means *(State of affairs)*:	The means by which a commercial transaction occurs.
Money:	Money is the thing given in exchange for Goods in a transaction.
Period_of_iteration:	The length of time from when the commerce event began to be repeated to when it stopped.
Place *(Locative relation)*:	Where the event takes place.
Purpose *(State of affairs)*:	The purpose for which an intentional act is performed.
Purpose_of_goods:	The the Buyer's intended purpose for the Goods.
Rate:	In some cases, price or payment is described per unit of Goods.
Reason *(State of affairs)*:	The Reason for which an event occurs.
Recipient:	The individual intended by the Buyer to receive the Goods.
Seller *(Source)*:	The Seller has possession of the Goods and exchanges them for Money from a Buyer.
Time *(Time)*:	When the event occurs.
Unit:	This FE is any unit in which goods or services can be measured. Generally, it occurs in a by-PP.

Lexical Units

buy.v, purchase.n, purchase.v

Figure 4.2: The Commerce_buy Frame, adapted from the FrameNet corpus.

4. I **purchased** the calculator for easier calculation of my debts.

5. Jon **bought** some expensive apples at five dollars a pound!

6. You **bought** me three pairs already!

7. Lee **buys** potatoes by the pound.

4.3 Scripts

Scripts, sometimes also called *schemata*, encode conventionalized knowledge about processes and the sub-processes they consist of. These processes could be actions, events, etc. A well-known example for this is Schank and Abelson [1977]'s restaurant script (see Figure 4.3), presented here in a slightly adapted version [Endres-Niggemeyer, 1998, p.29]. The script describes the typical events involved in having a meal at a restaurant. (Note that what is considered "typical" is subject to cultural and regional differences, among others.) The conventionalized process is subdivided into four *scenes*: entering the restaurant, ordering food, eating the foot, and paying the check and leaving. Each of the scenes can be broken down into even more basic actions.

Scripts share several commonalities with Semantic Frames, although their primary purpose is knowledge representation and not to be a theory of linguistic semantics. But both allow for the description of situations, and both provide facilities to model complex scenes out of simpler scenes (see also Section 4.4 below). We will leverage these commonalities, more specifically that Frames can be used to encode script knowledge, when transforming meeting interpretations to summary representations in Chapter 5.5.

4.4 Corpora

FrameNet

FrameNet [Fillmore and Baker, 2010] is a large-scale research effort to apply Fillmore's theory of Frame semantics to real-life data by annotating substantial parts of existing corpora, such as, the British National Corpus [BNC07]. There are also a number of sister projects for other languages than English, such as, German FrameNet [Boas et al., 2006] and SALSA [Burchardt et al., 2006] as well as Spanish FrameNet [Subirats and Petruck, 2003].

The data provided by FrameNet consists of two parts, Frame definitions and corpus annotations. The project has defined over 1000 distinct Frames in the current release 1.5, each of which consists of a general description, definitions of Frame

name: restaurant

props:
tables
menu
food
bill
money

entry conditions:
customer is hungry
customer has money

roles:
customer
waiter
cook
cashier
owner

results:
customer has less money
owner has more money
customer is not hungry

scene 1: ***entering***
customer enters restaurant
customer looks for table
customer decides where to sit
customer goes to table
customer sits down

scene 2: ***ordering***
customer picks up menu
customer looks at menu
customer decides on food
customer signals waitress
waitress comes to table
customer orders food
waitress goes to cook
waitress gives food order to cook
cook prepares food

scene 3: ***eating***
cook gives food to waitress
waitress brings food to customer
customer eats food

scene 4: ***exiting***
waitress writes bill
waitress goes over to customer
waitress gives bill to customer
customer gives tip to waitress
customer goes to cashier
customer gives money to cashier
customer leaves restaurant

Figure 4.3: The restaurant script.

Elements, and a (not necessarily complete) list of lexical units known to evoke the particular Frame.

Frame Elements can be characterized by how central they are to the Frame in which they appear. FRAMENET distinguishes between four types of levels:

- Core
- Core-unexpressed
- Peripheral

- Extra-thematic

A *core* FE denotes a “conceptually necessary component of a frame, while making the frame unique and different from other frames” [Ruppenhofer et al., 2006, p26]. For example, in the COMMERCE_BUY Frame, the FE’s BUYER and GOODS have *core* status.

FRAMENET provides an inheritance hierarchy of Frames (see Section 4.4). This notion includes that Frames further down in the hierarchy inherit the Frame Elements from their super-Frames. However, in some cases the lexical units that evoke a particular Frame already inherently encode what the super-Frame expects to be a distinct Frame Element. Consider for example the following two sentences:

1. I’ll **do** the exercise later.

2. I’ll **practice** later.

In the first sentence, *do* evokes the very general INTENTIONALLY_ACT Frame. This Frame defines a *core* FE ACT, which in this particular sentence is filled by *the exercise.* In the second sentence *practice* evokes a sub-Frame of INTENTIONALLY_ACT, namely PRACTICE. Conceptually, that Frame contains an *act*, too, but it is already clear from the evoking LU that the act has to be some kind of practice. For that reason PRACTICE does not have to inherit ACT from INTENTIONALLY_ACT. Such FE’s are labeled as *core-unexpressed.*

In contrast with *core* FE’s, some Frame Elements, such as, TIME, PLACE, DEGREE etc. do not add to the distinct description of a framal situation. They are general enough that they may appear in many different Frames that otherwise share no or hardly any other similarity, i.e., despite adding to the situational description of the Frame, they are mostly independent of the Frame. Such Frame Elements are called *peripheral* for that reason.

A similar case, in which certain Frame Elements do not have *coreness* status, are *extra-thematic* Frame Elements. They differ from peripheral FE’s in that they are not completely independent from the Frame in which they appear. They describe framal situations themselves that overlap with their host Frame. For instance, a Frame Element DURATION appear across a large range of otherwise independent Frames in FRAMENET. But a duration is always a duration *of something,* and that something typically overlaps with parts already covered by the host Frame. Other *extra-thematic* Frame Elements include FREQUENCY and ITERATION.

Some of the FE’s in FRAMENET expect their fillers to exhibit certain properties. For instance, Figure 4.2 asks for fillers of the Frame Element PLACE to be of the type *Locative relation,* and fillers of TIME to be of type *Time.* These specific kind of requirements are called *Semantic Types*; three types of semantic types are defined in

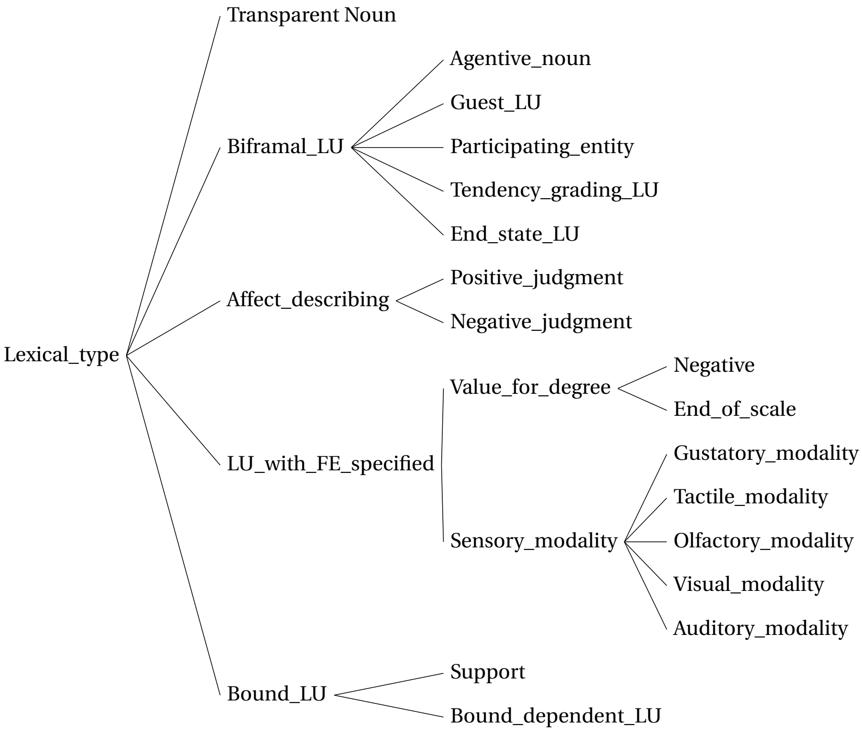


Figure 4.4: Lexical Types in FRAMENET

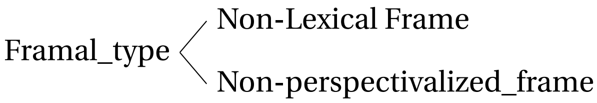


Figure 4.5: Framal Types in FRAMENET

FRAMENET: *lexical types* (see Figure 4.4) , *framal types* (see Figure 4.5), and *ontological types* (see Figure 4.6). In FRAMENET, lexical types add further information on lexical units, framal types on Frames, and ontological types on the filler entities of Frame Elements. In our approach we concentrate on the last type, see below.

Lexical and Framal Types

For a given lexical unit, the *lexical type* information further characterizes the LU in addition to the Frame it evokes. For instance, both the verb *compliment.v* and *scold.v* evoke the JUDGMENT_DIRECT_ADDRESS Frame, although their respective meanings are on rather opposite sides of the scale. That is the reason *compliment.v* is annotated with the lexical type *Positive_judgment* while *scold.v* is annotated with *Negative_judgment* in FRAMENET (cf. [Ruppenhofer et al., 2006, 112ff]).

Currently, FRAMENET defines two specific framal types, *Non-Lexical Frame* and *Non-perspectivalized_frame*. The former type denotes Frames that exist for structural reasons in the relation net of FRAMENET (see Section 4.4), but do not themselves contain any lexical units. Non-perspectivalized Frames on the other hand may contain a large number of quite diverse lexical units. These lexical units all allude to the same background scene, but from varying perspectives. A Frame marked as non-perspectivalized could potentially be split up in more specific sub-Frames which take on these different perspectives.

Ontological Types

Ontology is the classic philosophical discipline studying *being* and *existence*. The term was introduced at the beginning of the 17th century by Lorhard [1606], and the field is generally seen as a sub-field of metaphysics. One of its foremost tasks is that of determining categories that all *being* can be classified into, where a category groups members with common properties. We can trace discussion of these questions back to the ancient Greeks. For instance, in [Aristotle, 2000], the author attempts to divide all *being* into ten distinct categories, which he derives in parts from linguistic observation. His works set the grounds for over 2,000 years of philosophical discourse.

Today, this discipline has gained popularity outside Philosophy too, for instance, in such fields as Cognitive Science and Artificial Intelligence for modeling system knowledge about the world. Concrete models of knowledge that follow ontological principals, i.e. specific categorizations of entities into concepts and relations between them, are often called "ontologies" themselves.[3] Gruber [1993] defines an ontology as an "explicit specification of a conceptualization".

The ontological types in FRAMENET are based on this notion of "ontology". They provide a categorization of entities. The categories are ordered in a hierarchy, where the relation between a parent node and a child node is *subsumption* of categories, i.e., every entity that belongs to a certain category also belongs to that category's parent category.[4] On the top-most level, there is a distinction into five basic categories:

Attribute Instances of this category are typically dependent on some other instance; they describe some property of that instance.

Physical Entity Physical entities are entities that can be placed in a simple four-dimensional world-view, where width, height, depth and time are the four

[3] To distinguish between the discipline and a concrete technical world model, we will write the discipline with a capital 'O' and refer to the model will all lower-case letters.

[4] In the context of ontologies, the subsumption relation is sometimes called *is-a*.

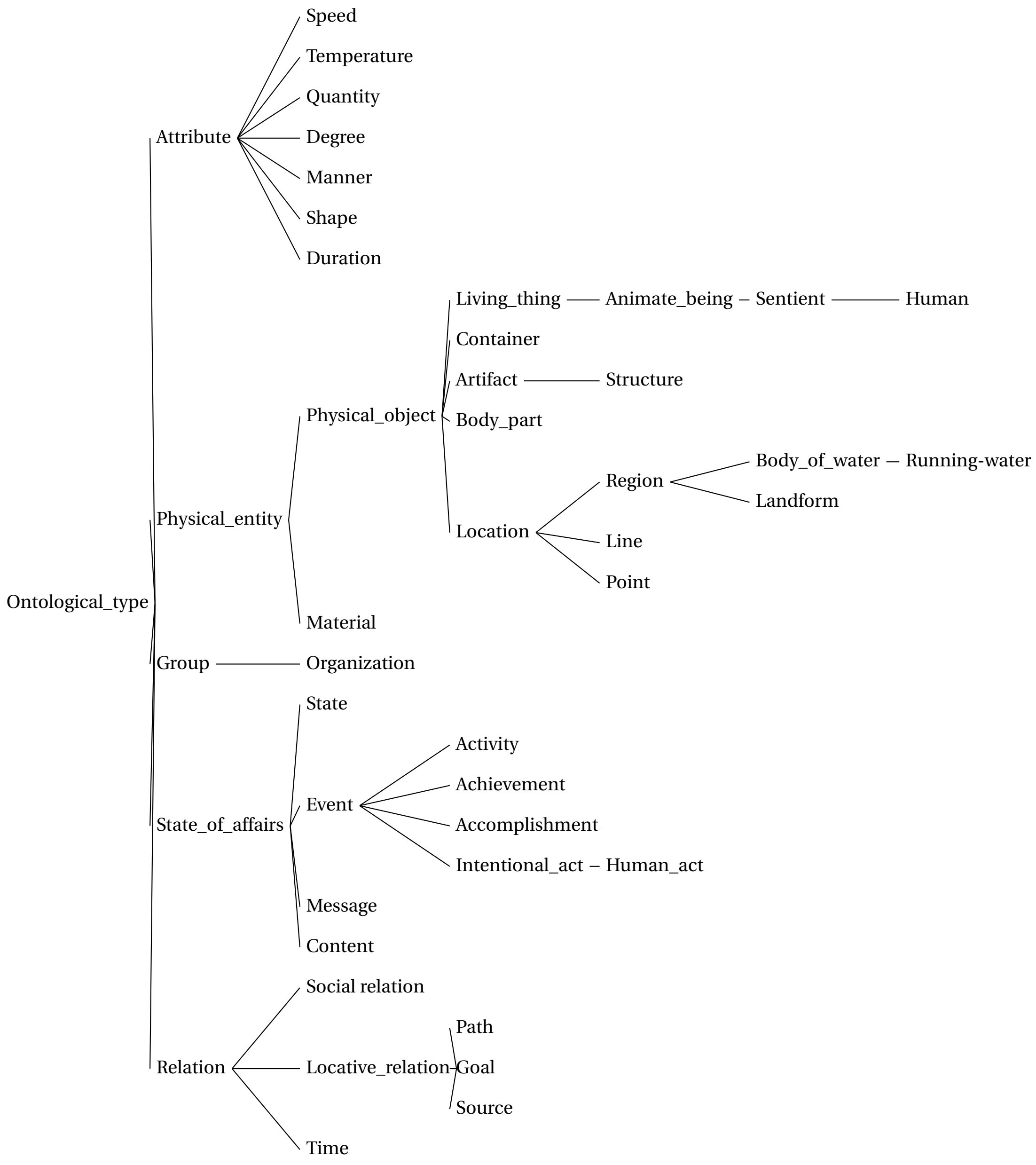


Figure 4.6: Ontological Types in FRAMENET

dimensions. On more formal grounds, this category can be understood to refer to a certain sub-group of *endurants* Grenon [2003]; Masolo et al. [2003].

Group This category refers to groupings of things, most typically Physical Entities, and more specifically people, reifying them as singular entities.

State of Affairs This category fuses non-physical *endurants* and *perdurants*.

Relation Relations provide specifications to integrate entities into State of Affairs.

All other ontological types fall under one of these five top-level categories, for instance TIME *is-a* RELATION.

It can be argued that the hierarchy of ontological types does not constitute an ontology in the strictest sense, but rather a *taxonomy*, the difference being that an ontology typically defines further relations between the categories. The additional axioms restrain the possible interpretations of the terminological symbols defined in the taxonomy of an ontology.

For practical purposes, semantic types can be used as a sanity check in Frame analysis. For example, if we want to analyze a given piece of text as evoking the COMMERCE_BUY Frame, but what we intend to annotate as *Seller* is known not to be a SOURCE, we can deduce that our analyses must be flawed. In turn, semantic types can also build the basis for symbolic inference. For instance, when handed a valid Frame analysis of a piece of text, we can infer that the entity referred to by the Frame Element *Seller* must be of the ontological type SOURCE, and thus have all the properties we associate with that category.

Frame Relations

The net that the project's name alludes to emerges from inter-Frame relations.

Similar to the hierarchical arrangement of semantic types, Frames themselves are structured in a subsumption or *inheritance* hierarchy. For instance, the COMMERCE_BUY Frame is a specialization of the more general GETTING Frame. Subsumed Frames inherit the Frame Elements of their ancestor Frames, however, like the Frame itself, the contained Frame Elements may get specialized too. For instance, GETTING's *Recipient* FE is specialized into *Buyer* in COMMERCE_BUY. However, a subsumed Frame may add its own Frame Elements that have no correspondence in the Frame it inherits from.

A similar Frame-to-Frame relation that should not be confused, with inheritance is the *sub-Frame* relation. When a Frame describes a rather complex situation in which certain parts may itself be described as Frames in their own right, these Frames are called sub-Frames of the encompassing Frame (the super-Frame). Similar to the inheritance relation, certain Frame Elements of the super-Frame may be mapped to FE's of the sub-Frames.

It can be useful not only to divide a complex Frame into sub-Frames, but also to order these sub-Frames by a temporal ordering relation. In FrameNet, this relation is called *precedes*. Ordering sub-Frames results in a richer knowledge-structure that can be beneficial especially in discourse interpretation when inferences in the spirit of Schank's scripts should be drawn, or when the *Construction* macro-rule should be applied in the transformation phase (cf. Chapter 3).

Situations can sometimes be seen from different points of view. For a Frame describing such a situation, the points of view may themselves be described by Frames in their own right. For instance, the Commerce_buy Frame is a *perspective_on* the more neutral Commerce_goods-transfer which in turn is a non-lexical sub-Frame of the Commercial_transaction Frame.

Usually, there is not only a single perspective on a neutral Frame, but at least two (or more). In our example, the Commerce_sell Frame presents a second perspective on Commerce_goods-transfer.

The *causative_of* relation hold between two Frames, if one of the Frames is the outcome of the other Frame. Similarly, the *inchoative_of* relations refers to a situation in which one Frame describes the beginning of another Frame. The two relations can be illustrated with the following chain between the Frames Attaching, Inchoative_attaching, and Being_attached.

$$\begin{tikzcd}
\textsc{Attaching} \arrow[d, "\textit{causative\_of}"] \\
\textsc{Inchoative\_attaching} \arrow[d, "\textit{inchoative\_of}"] \\
\textsc{Being\_attached}
\end{tikzcd}$$

Attaching is a Frame that describes the situation where an agent actively causes an item to attach to a certain goal location. Inchoative_attaching describes the process in which two or more items come to be attached to each other. Being_attached describes a state in which such items *are* attached to each other.

The *using* relation denotes a general semantic relationship between two Frames that is none of the aforementioned relations.

The *see_also* relation is mentioned here only for the sake of completeness, as it is not so much a semantic link, but a hint for users of FrameNet, e.g. annotators. When certain Frames are in some way closely related *see_also* may document such a relation to facilitate working with FrameNet. It's purpose is for the user to differentiate, compare, or contrast two (or more) Frames such related.

Table 4.1: Frame-to-Frame relations in FRAMENET

Inheritance	Specialization of Frames
Sub-Frames	Description of partial events of a Frame
Precedes	Temporal ordering of Frames
Perspective_on	Different points of view on a more neutral Frame
Causative_of	Outcome of a Frame
Inchoative_of	Beginning of a Frame
Using	Residue class of semantic relations.
See_also	For documentation purposes.

Design Meetings and the AMI Meeting Corpus

The Augmented Multi-party Interaction Project (AMI) lasted from 2004 until 2006 and was sponsored by the European Union under the Sixth Framework Programme with an overall budget of 8.8 million Euros. 14 partners from universities, research institutes and the industry in seven different countries participated in the project. The objectives of the project were to study meetings and how technology could be used to improve them. The range of studies was manifold, from multi-channel signal processing and automatic speech recognition to developing methods to detect different topics in a meeting or dominance patterns of the participants.

One of the main efforts of AMI was the development of a richly annotated meeting corpus, henceforth "the AMI corpus", which builds the base for the research reported in this thesis. It is freely available at `http://corpus.amiproject.org`.

Meeting Recordings

The AMI corpus consists of 171 recorded meetings. A meeting lasts 33 minutes on average and typically has four participants. All meetings are held in English although not all of the meeting participants are native speakers of English.

The meetings were recorded at three different sites in specially equipped meeting rooms. From the 171 meetings, 76 were recorded at the University of Edinburgh, 40 at the Nederlandse Organisatie voor Toegepast Natuurwetenschappelijk Onderzoek (TNO) in the Netherlands and 55 at the Idiap Research Institute, Switzerland. All meeting rooms used multiple cameras and microphones in parallel to record a meeting (a) from many different camera angles and (b) with multiple audio channels of varying quality. This was deliberately chosen to provide a rich database for computer vision and audio signal processing research. However, the concrete setup of microphones and cameras was different at the different sites, for instance microphones and cameras were installed at different angles. Figure 4.7 shows still cap-

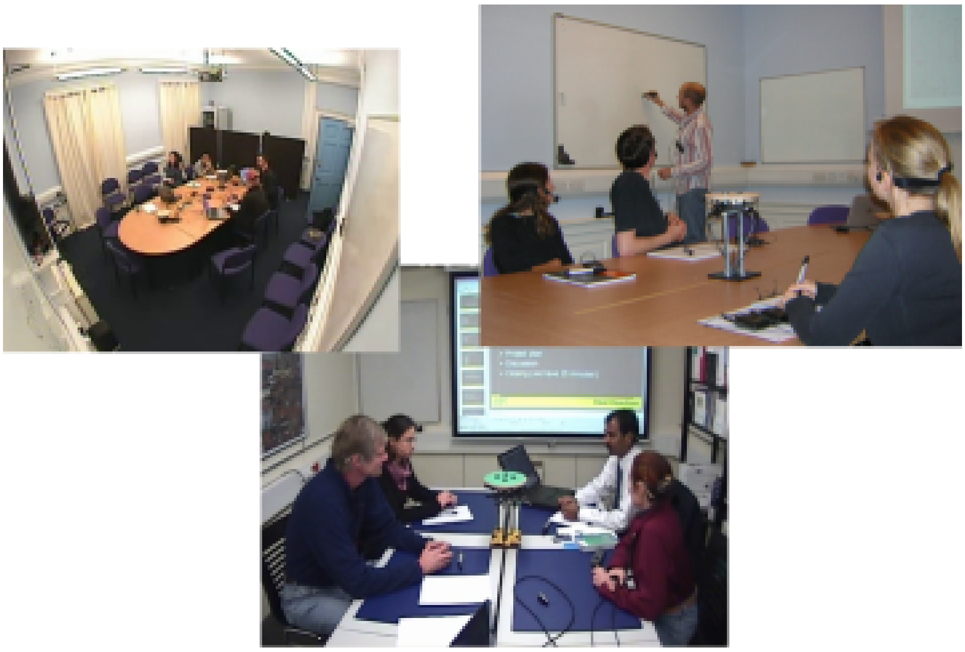


Figure 4.7: Stills from the AMI corpus showing the three different meeting rooms at University of Edinburgh, TNO and IDIAP (clock-wise from top-left).

tures from the overview cameras of all three sites.

The Remote Control Design Scenario

In 138 meetings, the four participants were given special instructions. The rationale was that the contents of roughly two thirds of the corpus should be restricted to a specific domain. To constrain the discourse of the meetings, a *scenario kit* was developed that defined the scenario of a virtual company. The four meeting participants were instructed that they were a team of employees.

This company—"Real Reactions"—is supposed to be producing remote controls for TV. A new project was just started and it is the task of the meeting team to design a new remote control. For this task, the four participants were assigned different roles they supposedly played in that company. One of them was the *project manager*, one was a *marketing expert*, another one a *user interface designer* and finally the last one was to be a *industrial designer*.

With these roles defined, the team had to attend four different meetings for the newly assigned design project. To begin their project, they hold a *kickoff-meeting* in which they introduced themselves and in which the project manager was to instruct everyone about the upcoming project. For this and other tasks, the scenario kit provides extensive side material that the four participants may use in the meetings. For instance, slide presentations were prepared ahead of time showing the agenda points of each meeting and other relevant contents. They were also given especially prepared laptops which they could use in and between meetings.

Other than that, the meetings were not pre-scripted, i.e., everyone in the meeting room is free to do and say what they like. As a consequence, in some meetings one can see the participants fall "out of role", but most of the time the participants were serious about their task.

When a meeting is over, the group splits and all four of the participants go to separate places to work on the task assigned to them during the meeting and to prepare for the next meeting. During these *individual work* session, the virtual scenario is continued to be simulated. For example, some participants receive an email from the Real Reaction's management with new project related information or they can conduct a marketing study (virtually) and receive the results of that study a little later.

The initial meeting of the group (the *kickoff* meeting) is mostly spent by the group to introduce themselves to each other and to familiarize with the technology available in the meeting room. The project manager then introduces the overall project and describes the division of work in this project according to the participants' roles.

After the kickoff meeting, the participants use their individual work phases to prepare for the second meeting. During this phase, the project manager is informed via (simulated) email that the budget for the project was cut by the management. The marketing expert receives result from a study about user requirements and desires. The user interface designer uses examples found on the simulated world wide web to devise an initial remote control design. In a similar fashion, the industrial designer devises functionalities, also on the basis of example from the web.

The second meeting is a *functional design* meeting in which the participants presented to each other the results of the individual work sessions in the form of slide presentations. In the following discussions they agree on the functional aspects of the remote control design.

In the individual work sessions after the second meeting, the project manager received an email informing her about deadline changes. The marketing expert finds out that fruit-based themes are in fashion for electronic devices and yellow is a popular color, guiding the group toward a banana-shaped remote control. The user interface designer collects examples of existing remote controls that may serve as examples for their own design. Likewise, the industrial designer collects information on components, properties and materials to use for the remote control.

The third meeting is a called the *conceptual design* meeting. Here, the participants report on their findings using the slide shows they prepared. In the discussion phase, they try to reach agreement on the conceptual design, also dealing with the changing project constraints and newly gained insights into the consumer market.

In the following individual work phase, the previous budget cuts are retracted to a certain degree and the project manager is informed about it via email. To evalu-

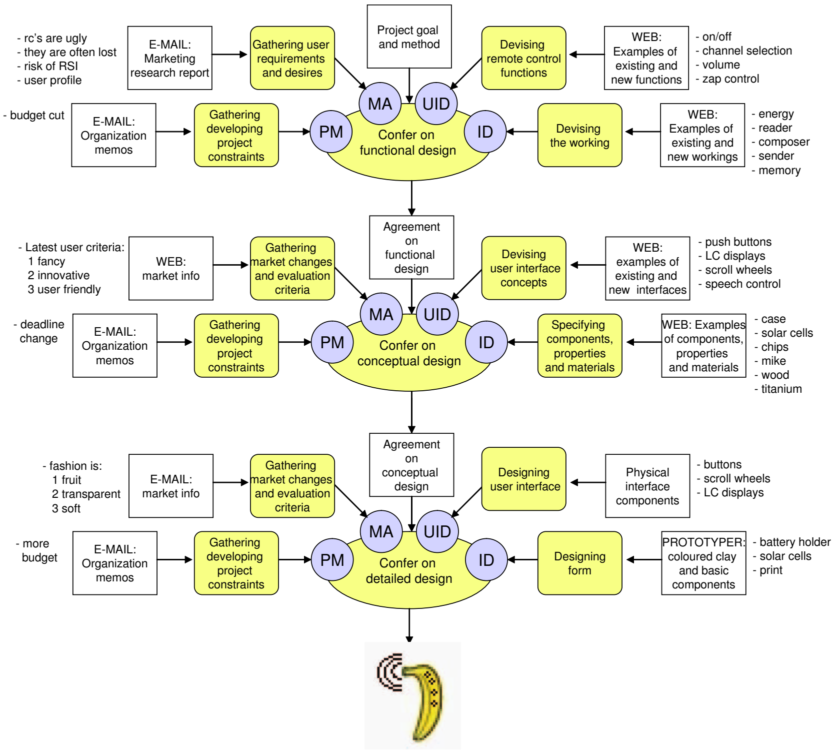

- rc's are ugly
- they are often lost
- risk of RSI
- user profile
E-MAIL: Marketing research report
Gathering user requirements and desires
Project goal and method
Devising remote control functions
WEB: Examples of existing and new functions
- on/off
- channel selection
- volume
- zap control
- budget cut
E-MAIL: Organization memos
Gathering developing project constraints
MA
UID
PM
ID
Confer on functional design
Devising the working
WEB: Examples of existing and new workings
- energy
- reader
- composer
- sender
- memory
Agreement on functional design
- Latest user criteria:
1 fancy
2 innovative
3 user friendly
WEB: market info
Gathering market changes and evaluation criteria
Devising user interface concepts
WEB: examples of existing and new interfaces
- push buttons
- LC displays
- scroll wheels
- speech control
- deadline change
E-MAIL: Organization memos
Gathering developing project constraints
MA
UID
PM
ID
Confer on conceptual design
Specifying components, properties and materials
WEB: Examples of components, properties and materials
- case
- solar cells
- chips
- mike
- wood
- titanium
Agreement on conceptual design
- fashion is:
1 fruit
2 transparent
3 soft
E-MAIL: market info
Gathering market changes and evaluation criteria
Designing user interface
Physical interface components
- buttons
- scroll wheels
- LC displays
- more budget
E-MAIL: Organization memos
Gathering developing project constraints
MA
UID
PM
ID
Confer on detailed design
Designing form
PROTOTYPER: coloured clay and basic components
- battery holder
- solar cells
- print

Figure 4.8: The three main stages of the AMI remote control design scenario after the kickoff meeting.

ate their work, the marketing expert develops an evaluation scheme, while the user interface and industrial designer cooperate to build a first prototype of the remote control, made of clay.

In the final meeting of the series, the *detailed design* meeting, the clay prototype is presented and discussed by the team. The evaluation scheme developed by the marketing expert is used to critically assess work on the prototype.

Figure 4.8 illustrates the sequence of the four meetings of the scenario kit with intermediate individual work sessions.

Storage and Retrieval

Recording the meetings produced a considerable amount of low-level data from microphones and cameras. It is therefore desirable to possess an infrastructure that facilitates access to these data. Synchronization of multiple channels is one issue, rich annotation of low-level data another one to be addressed.

The AMI corpus is encoded in NXT[5] which consists of two parts, a general corpus description language and a collection of tools to access, alter, and maintain a corpus encoded in this description language. The latter is based on an abstract object model [Evert et al., 2003] that allows to describe a corpus as a set of "observations" and the data collection attached to observations through a concept of layers.

There are three kinds of layers in NXT to encode different types of information. *Time-aligned* layers contain explicit temporal information for each of their contained elements. *Structural layers* contain element which are linked to other elements from which they inherit their temporal information. Elements that do not contain any time information at all, are placed on *featural layers*. They still may encode relations to other elements in the corpus through special pointer structures. Pointers together with the inheritance relations of structural layers form the backbone for creating rich dependency structures in NXT.

NXT corpus data can be serialized in XML. The definition of the corpus layout, the different layers, external resources and the actual observations are stored in a meta-data file. In addition to these abstract descriptions, the meta-data file also contains the relevant path information where the corpus data is stored. Data are highly modularized and distributed over several XML files. This way, users interested in specific features of the corpus can load only the data they are interested in without the overhead of loading irrelevant data. NXT includes an open-source Java-based API to access corpus data efficiently[6].

Corpus Annotations

In addition to the audio and video files, the AMI corpus provides a number of annotations. Some of these annotations are based on others, for instance "topic segments" are based on reference "words" (from the transcript) which are in turn synchronized with the timings of the audio/video signals.

All of the 171 meetings were transcribed by hand, using the Transcriber tool [Barras et al., 2001]. The transcripts were synchronized with the timings of the signal files using forced alignment. Not only the words uttered by meeting participants were transcribed, but were possible punctuation was inserted, too. Also, non-words sounds, such as, "laugh", "sigh", "sneeze", etc. were annotated in the transcript. In addition to manual transcription, different version of transcripts produced by the project's specifically developed automatic speech recognizer [Hain et al., 2007b] are also included in the corpus.

Besides the transcription, a rich variety of other annotations are available in the corpus, although not all of meetings were annotated with all the different annotation schemes. Table 4.2 gives an overview of the available annotations and Figure

[5] Nite XML Toolkit, `http://groups.inf.ed.ac.uk/nxt/`

[6] http://sourceforge.net/projects/nite/

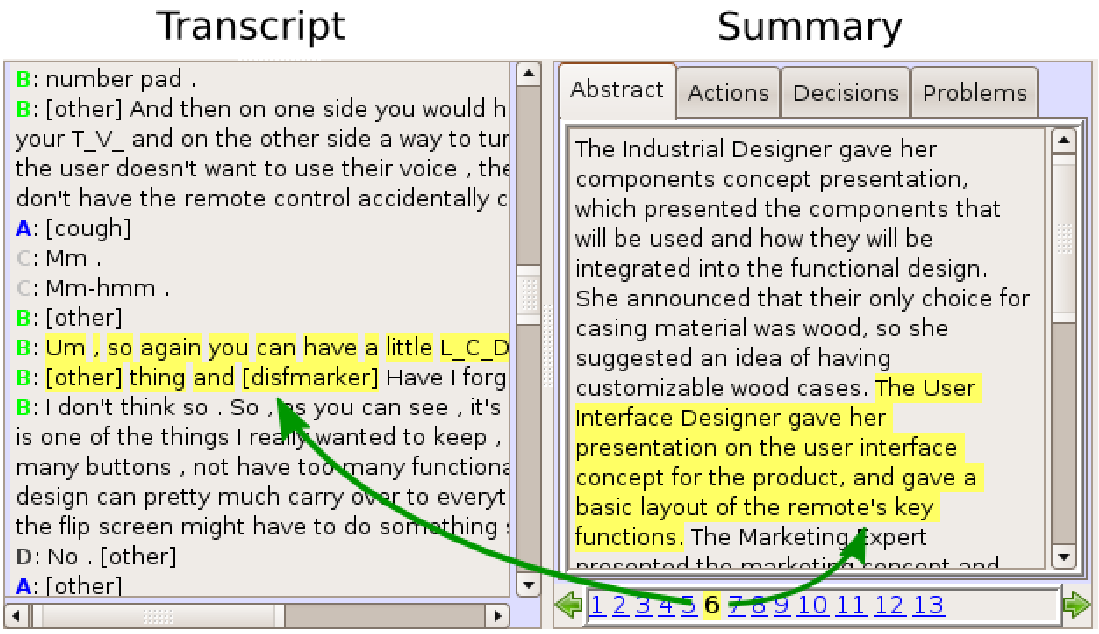
Table 4.2: The different annotation layers available in the AMI corpus

Layer	#Meetings	Description
ASR	169	Automatic speech recognition
ASRsubwords	169	ASR subwords (e.g. `isn't` → `is n't`)
abstractive	142	Manual abstractive summaries
chunks	4	Syntactic chunks
decision	47	Segments containing decisions
dialogueActs	139	Dialog acts
disfluency	138	Speech disfluency information
dominance	36	Speaker dominance
extractive	137	Manual extractive summaries
focus	14	Focus of attention of participants
handGesture	17	Hand gestures
headGesture	46	Head gestures
movement	125	Position of participants in the room
namedEntities	117	Named entities
participantSummaries	89	Summaries written by participants
segments	171	Manual speech segments
spurts	169	Automatic pause-based speech segments
subjectivity	20	Annotation of subjectivity
subwords	171	Manual subwords (see ASRsubwords)
topics	148	Predefined meeting topics
wordAlignment	169	Transcript / signal alignment
words	171	Manual transcript

4.10 illustrates a number of different annotations for a specific meeting, synchronized to a time-line, shown in a special NXT-based corpus viewer tool.

Naturally, the included hand-written abstractive summary annotations are of particular interest for us. Structurally they are split into four sections: an abstract that summarizes the meeting as a whole, a list of action items the meeting participants assign to each other during the meetings, the decisions made during the meetings, and any problems that the group encounters during the meeting. Only the abstract is a continuous text, the other three sections are usually recorded in form of bulleted lists.

This form of structured summary is somewhat peculiar. While the first part, the abstract, resembles the traditional understanding of a hand-written summary it typically does not contain the decisions, action items and problems of a meeting, as these points have their own separate sections. Consequently, there is no single coherent, all-encompassing summary text available for a meeting. The reason for this structure lies in AMI extensive annotation effort; in addition to the abstractive summaries, the corpus also encodes extractive summaries for each meeting. These are annotated by linking each sentence or bullet point in an abstractive summary to those utterances from the original meeting transcript that are understood to be supportive of the statement made in that sentence or bullet point. This mapping defines a binary extraction decision on the utterances in the meeting transcript from which an extractive summary can be reconstructed as follows: an utterance gets extracted if and only if it is linked into from four-fold summary. Figure 4.9 illustrates this mapping, where the concatenation of the selected utterances in the transcript form the extractive summary. Through the use of four distinct sections, extractive summarization systems can make use of a richer annotation, as the gold-standard extraction sentences are not only annotated as salient or not, but inherently contain the information to which of the four sections they contribute. This may be an informative feature for machine learning approaches, but also allows for new types of summarizers altogether that are more specialized to summarize only a particular



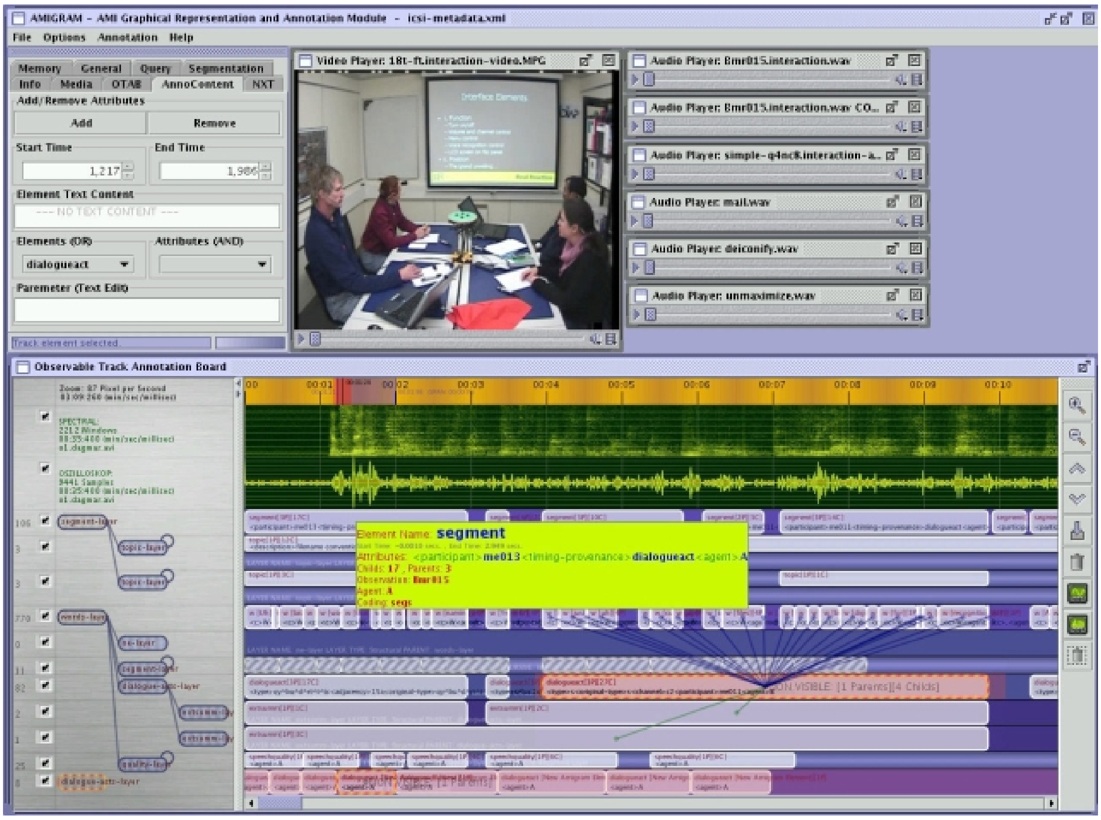
B: number pad .
B: [other] And then on one side you would h
your T_V_ and on the other side a way to tur
the user doesn't want to use their voice , the
don't have the remote control accidentally c
A: [cough]
C: Mm .
C: Mm-hmm .
B: [other]
B: Um , so again you can have a little L_C_D
B: [other] thing and [disfmarker] Have I forg
B: I don't think so . So , as you can see , it's
is one of the things I really wanted to keep ,
many buttons , not have too many functiona
design can pretty much carry over to everyt
the flip screen might have to do something
D: No . [other]
A: [other]

The Industrial Designer gave her components concept presentation, which presented the components that will be used and how they will be integrated into the functional design. She announced that their only choice for casing material was wood, so she suggested an idea of having customizable wood cases. The User Interface Designer gave her presentation on the user interface concept for the product, and gave a basic layout of the remote's key functions. The Marketing Expert presented the marketing concept and

Figure 4.9: Extractive summaries as links between transcript and abstractive summary in the AMI corpus. Here, the highlighted sentence in the meeting summary (right-hand side) links to 13 utterances in the meeting transcript (left-hand side). The 6th of these utterances is currently selected.

aspect of meetings. For instance, Hsueh [2008] concentrates on the detection of decisions in a meeting; while Purver et al. [2007] identify the action items attached to the participants during a meeting automatically.

At the same time, it would be straight-forward to construct a single text out of the four-fold structure, by inserting the decisions, actions and problems of the last sections into the text of the abstract section. The link mapping that is used for extractive summaries also provides a temporal layout of the contents of the four-fold summary, thereby giving the possible entry points where the bullet points could be inserted into the abstractive text. For a concise text style, a potentially shortened bullet point would also have to be transformed into a full sentence. Such a process could be useful, for instance, when looking for a gold-standard summary to evaluate the generated summaries against, where a more traditional form of a single piece of coherent text may be called for.



AMIGRAM - AMI Graphical Representation and Annotation Module - icsi-metadata.xml

Start Time: 1,217

End Time: 1,986

Element Text Content: --- NO TEXT CONTENT ---

Elements (OR): dialogueact

Track element selected.

Element Name: segment

Attributes: <participant> me013 <timing-provenance> dialogueact <agent> A

Childs: 17 , Parents: 3

Observation: Bmr015

Agent: A

Coding: segs

Figure 4.10: The AmiGram tool [Lauer et al., 2005] displaying temporally aligned annotations for a recorded AMI meeting

Besides the summary annotations, one other annotation layer is of particular interest because it touches on the question of content representation. While Frames are well-suited to capture the semantic content of utterances made during a meeting, not all aspects of the discourse can be described by Frames only. The conversations between the meeting participants are characterized by exchanging information, but also by requesting information, signaling understanding, being polite etc. Some of the utterances thus have an operational function in the discourse, rather than providing the other participants with propositional information.

Alexandersson [2003] argues, that for the summarization of dialogs, such effects have to be taken into account. His representation formalism (see also Section 3) thus combines propositions with a representation of *dialog acts* (DA's) [Bunt, 1994]. This notion derives from advancements of Austin's original speech act theory [Austin, 1975]. The underlying idea is to describe the basic *communicative function* of utterances with respect to the discourse. This includes distinctions such as whether an utterance is a question or a statement, but a particular implementation may also model more fine-grained differentiations such as whether a question is a Yes/No-question or a Wh-question etc. There is no single generally agreed-upon set of dialog acts, although a number of different ones have been proposed and used on different corpora (e.g., SWITCHBOARD [Jurafsky et al., 1997], VERBMOBIL [Alexandersson et al., 1997] ICSI MRDA [Shriberg et al., 2004], MALTUS [Popescu-Belis, 2004]).

In the AMI corpus, the dialog act annotation is based on a tag set of 15 distinct dialog act labels. These are:

Assess The speaker judges a current topic of the discourse, or offers an opinion on it.

Backchannel The speaker signals that he or she is following the content of the utterances of another speaker.

Be-Negative The speaker says something impolite or negative.

Be-Positive The speaker says something polite or positive, e.g.. in a fixed phrase such as a greeting.

Comment-About-Understanding The speaker explicitly says that he or she understands what is currently talked about in the discourse.

Elicit-Assessment The speaker asks one or more of the other speakers to judge a certain topic, or to tell his or her opinion on the topic.

Elicit-Comment-Understanding The speaker wants to reassure him- or herself that another speaker or a set of speakers is able to follow along.

Elicit-Inform The speaker asks another speaker to make a statement on a certain topic of discourse.

Elicit-Offer-Or-Suggestion The speaker asks another speaker or multiple speakers to make a suggestion or offer.

Fragment The speaker starts to say something, but breaks off the utterance before the full sense of it could be determined.

Inform The speaker makes a statement about something, thereby informing the listeners about a topic of discourse.

Offer The speaker offers to say or do something.

Stall The speaker makes an unintelligible noise.

Suggest The speaker makes a suggestion or proposal.

The annotation of dialog acts can be divided in two sub-tasks: segmenting a continuous stretch of speech into sensible units and labeling each of these segments with one of the dialog act tags. The AMI corpus provides manual and automatic annotation for both of these sub-tasks. Dialog act segments have an important function besides carrying the DA label: Since full grammatical sentences are rare in spontaneously spoken discourse, as discussed before, DA segments subdivide long speaker contributions into sentence-like units. This fact has been utilized by the aforementioned summary link annotation: here, links connect the actual sentences in the abstractive summary with the sentence approximation provided by dialog act segments in the transcript.

Figure 4.11 shows the distribution of dialog acts in the AMI corpus. The red bars refer to all dialog act segments in the corpus, while the green bars only count the occurrences of the different dialog acts in those segments that are linked by their meeting's respective summary. Interestingly, there is a notable difference for the distribution of *statement* DA's, which will be of importance for us. It implies that the dialog act label can be an important source of information in determining which parts of the discourse are relevant.

In addition to the labeling of dialog act segments, a link annotation is available for some of the recorded meetings, that encoded so-called *adjacency pairs*, i.e., pairs of dialog acts that refer to each other as, for instance, question and answer.

To make annotations available to an end-user, be it for the inspection purposes or for adding new annotations, NXT contains a rich component library based on its Java API. These re-usable components facilitate the implementation of new annotation tools. In fact, a number of such tools are available for the AMI corpus, for example, to play back audio and video data, to display a transcript, etc.

The AMIDA Corpus

The AMIDA project is a successor to the AMI project, continuing research focusing on meetings. In comparison to AMI, AMIDA changes the main focus of research from archived meeting recordings to online support for meeting participants, and

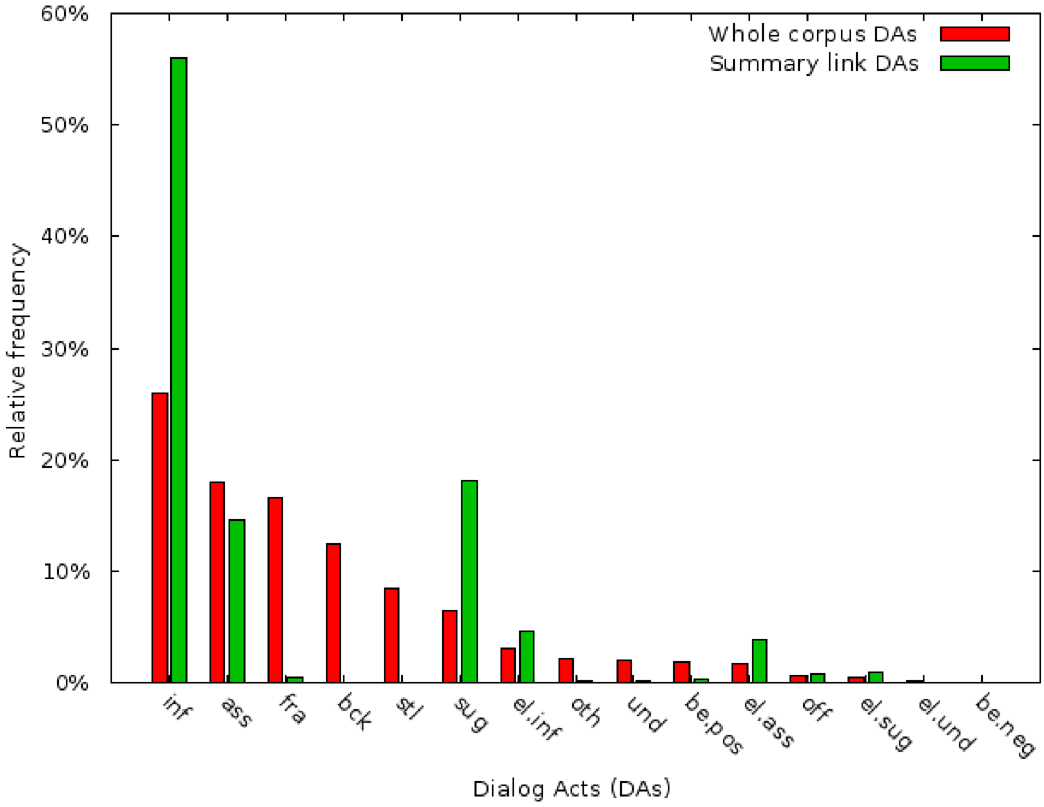


Figure 4.11: Distribution of Dialog Acts in the AMI Corpus.

the addition of participants remotely connected via video and/or audio conferencing technology.

To provide comparability, the background scenario (designing a TV remote control) is kept the same. In fact, the meeting groups in AMIDA meetings continue the work of an AMI group after the latter's second meeting (the *functional design meeting*). The participants' briefing establishes a rationale for this particular setup by informing the group that they are to take over and finish the work which another group in the company has begun. The new group has access to the recorded meetings and all other material of the old group through a meeting browser.

The first meeting of the AMIDA group corresponds to the *conceptual design meeting*. Before the meeting the participants get some time to familiarize themselves with the recorded meetings of the previous group.

The AMIDA meetings were recorded in a similar fashion to the AMI meetings. All in all, the AMIDA corpus consists of about 20 hours of recorded meetings, half of which are fully transcribed and annotated with the following annotation layers:

Table 4.3: The different annotation layers available in the AMIDA corpus

Layer	**#Meetings**	**Description**
addressing	13	Addressee information
adjacencyPairs	5	Pairs of referring dialog acts
dialogueActs	5	Dialog acts
disfluency	5	Speech disfluency information
extractive	5	Manual extractive summaries
handGesture	12	Movements of hands
headGesture	12	Movements of heads (nodding, etc.)
namedEntities	17	Named entities
segments	19	Manual speech segments
subjectivity	12	Annotation of subjectivity
topics	3	Predefined meeting topics
words	19	Manual transcript

4.5 Chapter Summary

This chapter introduced some necessary background knowledge and the theories used in this thesis. They are the basis for the abstractive meeting summarization approach laid out in the following chapter.

A central theory we make use of is that of Frame Semantics, which this chapter described in detail. Semantic Frames are used as the backbone for the representation of both meeting and summary contents.

Our approach is a data-driven approach, i.e. we stipulate our summarization method to deal with non-contrived, real-life data. We described the AMI corpus and AMIDA corpus which provide such data in form of recorded meetings in a specific scenario.

Chapter 5

The MEESU *system*

5.1 Introduction

In this chapter, we present the details of our own approach to abstractive meeting summarization. In Section 5.2 we give an overview of the general architecture of the meeting summarizer MEESU, introducing the different components and knowledge sources involved. In the subsequent sections (5.3–5.6) we explain the different parts of that architecture in greater detail.

While the ITG model presented in Section 3.2 is a general, abstract description of all summarization processes, its realization in an actual abstractive meeting summarizer is a novelty. Particular contributions we make to the field include a novel representation formalism for meeting contents, in which we show how Frame Semantics (Section 5.4) can be leveraged to represent the content of a meeting. Drawing on results from cognitive science, we then demonstrate that such a representation is suitable for macro-rules that transfer a meeting representation into a summary representation (Section 5.5). Two new auxiliary inference rules are introduced that facilitate the application of the macro-rules. Finally, we present a novel algorithm for the generation of English text from a Frame representation in Section 5.6, before summarizing this chapter in Section 5.8.

5.2 Architecture

The overall architecture of the MEESU summarization system implemented in the course of this thesis is shown in Figure 5.1. The left half of the schematic displays what is in most parts a classical pipeline architecture consisting of a pre-processing step followed by three phases, *interpretation*, *transformation*, and *generation*. The right hand side of Figure 5.1 shows the different knowledge bases used during processing. These phases follow closely the model by Spärck Jones described before. Each phase consists of a number of system components, visualized by cream-colored squares. The data that is passed between the different components is shown as green ellipses.

To summarize a given meeting, the first step in the pipeline is to make the meeting accessible to the system. For that, multimodal recordings of the meeting are

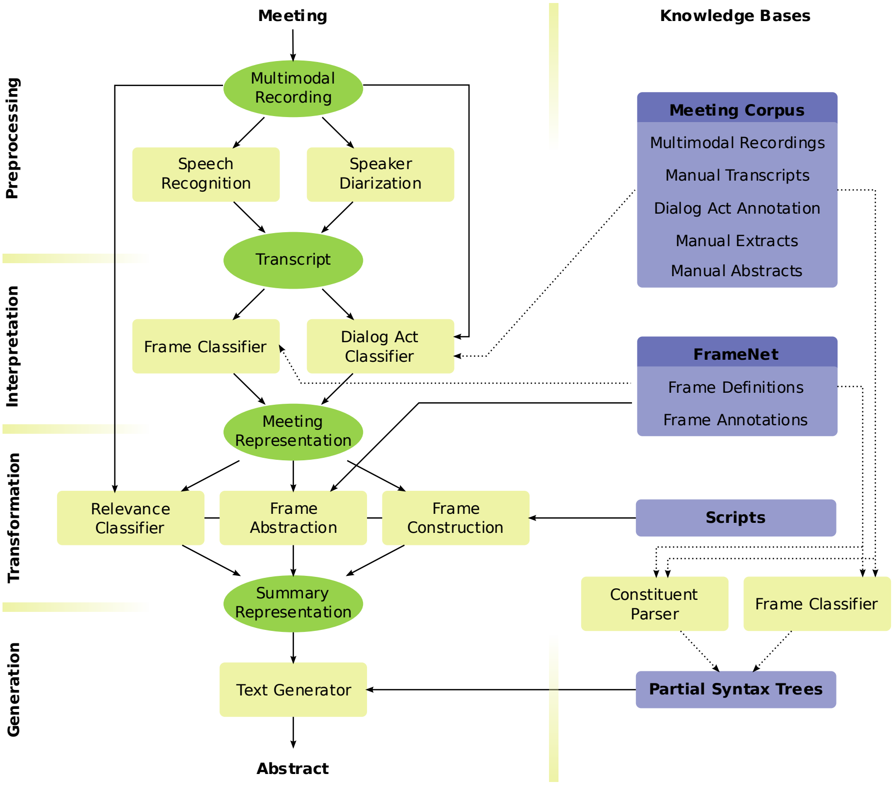


Figure 5.1: The architecture of the MEESU system. The run-time system is shown on the left, where the different sub-components are displayed as rectangles, and intermediate representations as ellipses. The arrows demonstrate the online information flow between the components. The right-hand side lists the used knowledge bases and which components are informed by them. Dashed arrows mark offline processing.

created. The result is a time-synchronized set of sound and video streams, potentially using multiple microphones and cameras, resulting in a rich signal selection.

The main source of input for the three main phases of the pipeline is a transcript of the meeting, produced from said recordings. In a fully automated system, the words that appear in such a transcript would be produced by automatic speech recognition. Sound and video features can be used to segment the recognized words, and to assign the segments to the different speakers in a process called *speaker diarization*. As an alternative, manual transcriptions provide a better quality, since automatic transcripts today still feature word error rates between 20 and 30 percent. But the meeting recordings are not only used to produce a transcript, some of the inherent features, such as pause information, are also used in the transformation phase (see below).

The interpretation phase consists of two main components, a Frame classifier and a dialog act labeler. These two components produce a semantic representation of the contents of the meeting. A dialog act captures the speaker's intent for every utterance made in the transcript, for instance, whether the speaker makes a *statement* on a certain content or asks a *question* about it. To determine such distinctions, the system offers two choices: manual dialog act annotations, if available, or automatic classification using a supervised classifier trained on the AMI corpus. The result is in both cases a dialog act label for each segment of the transcript.

For the detection of semantic Frames in the transcript, the SEMAFOR parser [Das and Smith, 2011] is used. It is a semantic role labeler that produces for each segment of the transcript a set of Frames evoked by different words in that segment, together with the corresponding Frame Element annotation. SEMAFOR relies on the FRAMENET database to create a probabilistic prediction model in an off-line training phase. This means that the model has to be learned only once, independent of a concrete meeting to summarize, and can then be re-used at run-time. For the final representation, the dialog acts produced by the dialog act classifier are mapped to Frames, too, to yield an integrated representation of both knowledge sources.

The transformation of the meeting representation into a summary representation is performed by three interdependent components. The relevance classifier decides on inclusion or deletion of parts of the meeting representation on the utterance level. This decision is exactly what extractive summarizers do, which is why we reuse an existing extractive meeting summarizer for this component [Murray, 2008].

Unlike extractive summarizers though, the system presented here contains two further components in the transformation phase. The Frame Abstraction component makes use of the Frame hierarchy defined in FRAMENET (see Section 5.4) to replace representational descriptions of certain concrete situations with more abstract descriptions. The rationale for this step is that the specificities of the concrete situation can be dropped to yield shorter descriptions. Another advantage is that multiple situations that are similar on an abstract level can be merged, thus reducing the size of the representation.

The Frame Construction component identifies sequences of Frames that together can be described by a single other Frame. For this, script knowledge informs this component about the sub-sequences more complex Frames are made of. If such a sequence is detected the members of the sequence are replaced by the complex alternative, thus reducing the representation size further. The script knowledge base has been created manually.

This phase in the system does not adhere to a pipeline architecture in the strictest sense, because the three components are run in turn multiple times until no further changes occur. The number of runs is thus not predetermined, but rather depends on the processing result of the components themselves.

The meeting representation is thus processed into a summary representation which is passed to a text generator. The main knowledge base for this generator is a set of partial syntax trees which define for the inventory of Semantic Frames how they can be mapped to syntactic structures. These trees are automatically extracted from corpus data in an offline pre-processing step, using again the SEMAFOR Frame classifier together with a English constituent parser [Klein and Manning, 2003]. The generator implements a search algorithm to combine multiple partial syntax trees, and thus multiple Semantic Frames, into a single sentence. The result is a textual abstractive summary of the meeting.

The following sections describe the underlying techniques for the three main parts of the pipeline in greater detail.

5.3 Preprocessing a Recorded Meeting

The first step in the pipeline is a process that transfers the contents of a meeting discourse from the raw signal into a representation formalism that is accessible to a computer. By "raw signal" we mean the lowest level of interface between the computer and the meeting in question, typically microphone (audio) and camera (video) signals that have been digitized.

Automatic Transcripts

While the AMI and AMIDA corpora both provide manual transcripts for every recorded meeting, an automatic summarization system cannot expect this in general. High-quality manual transcripts are laborious and costly to produce and if an automatic summarization system were dependent on the existence of such transcripts, we would find it difficult to argue why the same labor and costs could not be spent directly to create high-quality manual summaries. Thus we are looking for an alternative, automatic way to create meeting transcripts.

When people speak with each other, their utterances are transported through the air as sonic waves. Machines access the speaker utterances, e.g. for recording or live broadcasting to remote participants, through microphones which pick up the sonic waves and transform them into electromagnetic signals. In the first instance, these are analog signals. Digital computers need the analog signals transformed into digital signals in order to process them. In a process called *sampling*, the analog signals are probed at a certain rate. The sequence of resulting measurements of pitch and volume of the original analog signals can then be stored in digital form as a representation of the original sonic wave.

The quality of the recorded audio signal is essential for the quality of subsequent analyses. A number of factors influence that quality. On the production side, the ideal output is a clear, loud, constant signal against a silent background. In a real-life situation, this is typically not the case: cross-talking, noisy environments, and low talkers are examples that will lead to sub-optimal recording results. On the recording side, the quality (microphone type, transfer technology, etc.) and handling (microphone placement, leveling out, etc.) of the audio equipment determine the quality of the recording. Finally, the quality of the digitized signal depends on the hardware used to sample the analog signal as well as on the resolution of the discretization along the axes "time" and "amplitude".

Depending on the particular setup, the microphone used to record the voice of the speaker picks up more or less background noise. Close-talking microphones are thus often preferred as they typically convey a better ratio of signal to noise than far-field microphones, because they are closer located to the sound source. They also make it relatively easy to identify the current speaker, which is a non-trivial problem when using far field microphones. However, for the meeting participants, lapel microphones or headsets may feel obtrusive and unnatural. An alternative is thus to use microphone arrays which consist of multiple directional microphones that can be placed in a central spot in the meeting room. Multiple microphone recordings always bear the problem that the different audio tracks need to be synchronized accurately. However, through techniques such as beam-forming and digital filters they allow for a high quality multi-track recording that a single far field microphone could not deliver.

Speaker Diarization

Speaker Diarization is concerned with answering the question "who spoke when?". Given a recording of multiple speakers, the objective of a diarization system is to identify the different speakers in the recording and determine which parts of the recording were produced by which speaker. In this context, identification does not mean to reveal the real-word identity of a speaker, but rather the differentiation between multiple abstract candidates. A diarization system must be able to tell when a new speaker contributes to the discourse for the first time, and to recognize when a speaker has spoken before.

It is clear that speaker identification and diarization are important sub-tasks in generating meeting transcripts automatically. Without these disciplines, a transcript would merely be a stream of words without any speaker information attached. The result would be of questionable use, because the reader of such a transcript would find it difficult to follow speaker-related aspects, such as, the exchange of different points of view, task assignments, or addressing of other meeting participants. Reading a transcript that does not provide any speaker information at all would resemble reading the dialog of a theater play with all the role names removed.

Speech Recognition

Automatic Speech Recognition (ASR) analyzes the spoken dialog of a meeting and turns the audio recordings into text. It automatically transcribes what the participants said during a meeting.

In the state-of-the-art approach, an ASR system first segments the audio stream into signal and silence parts. Within the signal parts, a phoneme recognizer produces an estimation of a sequence of phonemes in the audio stream. The final step is to turn this phoneme sequence into words. This is a complex task which is typically achieved through a probabilistic search. For the probability estimation, not only the confidence of the phoneme recognizer is taken into account, but also acoustic and language models usually serves as a secondary sources of information.

A language model uses the fact that in natural conversations, some words are more or less likely to appear before or after certain other words. For instance, the combination “the would” is much less likely in English than “the wood”. This language-dependent knowledge can thus be of great importance for disambiguating between different word candidates.

Segmentation

The output of a modern ASR system is usually a list of the n-best recognition hypotheses, or a word lattice. For inclusion in a meeting transcript, such output is ultimately transformed into a sequence of word tokens. Even if the recording can be separated by speaker, either by using multiple speaker-specific microphones or through speaker diarization, the result would be one token sequence per speaker. But a separate transcript for each speaker makes it difficult for the reader to follow the flow of the meeting and synchronize exchanges between speaker, for instance, during a group discussion.

In a traditional transcript, we expect the different utterances to be interwoven to reflect the chronological order of the discourse. An automatic system should thus merge the transcribed speech of the meeting participants into one global transcript. This requires the identification of suitable positions at which a global transcript can change from one speaker to the next.

But even for passages in which only a single participant speaks, it may be desired to break the monologue into smaller parts to improve readability [Jones et al., 2003]. In particular, ASR systems do not recognize sentence boundaries or insert punctuation. Finding sentence boundaries in a stream of spontaneously spoken words is in fact a non-trivial task, since the speech material often does not even contain full sentences, but rather elliptical and ungrammatical expressions. Therefore, the notion of “sentence-like units” is used instead in some works, e.g. [Kolář, 2008].

Speech Disfluencies

An automatically produced transcript contains verbalizations of all sounds uttered by the speakers, including not only actual words, but also noise sounds, such as, filled pauses, coughs, sighs, laughs, grunts, etc. It is important for an ASR system to model such non-word noises, too, or else they would likely be mistaken for words or parts of words that the speaker never actually said.

For the reader of a transcript, such non-word sounds are often undesired as they interrupt the flow of words. For automatic systems, such as e.g. summarizers, which further process transcripts, non-word sounds complicate the analysis of the speech content. In addition, other artifacts of spontaneously produced speech find their way into the automatic transcript as well. Shriberg [1994] observes certain regularities with which *speech disfluencies* occur in human language production, allowing her to model them through acoustic and lexical features. For that, she develops an elaborate classification scheme for different types of disfluencies.

Shriberg's work is based on human-human and human-computer dialogs, as featured e.g. in the SWITCHBOARD Corpus [Godfrey et al., 1992]. Besser [2006] extends the work of Shriberg and others (notably Finkler [1997]) to the meeting domain, using data from the AMI Corpus. Besser's goal is to explicitly provide an annotation scheme (see Figure 5.2) that allows for the automatic correction of disfluencies to foster downstream processes.

Such an automatic system is introduced by Germesin [2008]. His GRODI system uses a hybrid approach that combines hand-written rewriting rules with statistical classification trained on Besser's data. GRODI detects and automatically corrects all disfluency types except *Disruptions* and *Uncorrected*. An evaluation showed that GRODI reduced the number of disfluencies in the test transcript by 50%, improving the amount of clean speech from 85.6% to 92.2%. Figure 5.3 shows an excerpt from AMI meeting ES2002a before and after cleaning the manual transcript with GRODI.

5.4 Interpretation of Meeting Contents

This section addresses two main questions:

1. What is a suitable representation formalism for subsequent summarization steps?
2. How can a given meeting be analyzed and mapped into this representation from raw signals?

At first glance, these questions may seem independent of each other, but in fact they are interlaced. When looking for a representation formalism in the context of an implementable system, we have to strive for a balance between expressiveness of the formalism on the one hand and practicability on the other hand. If the formalism

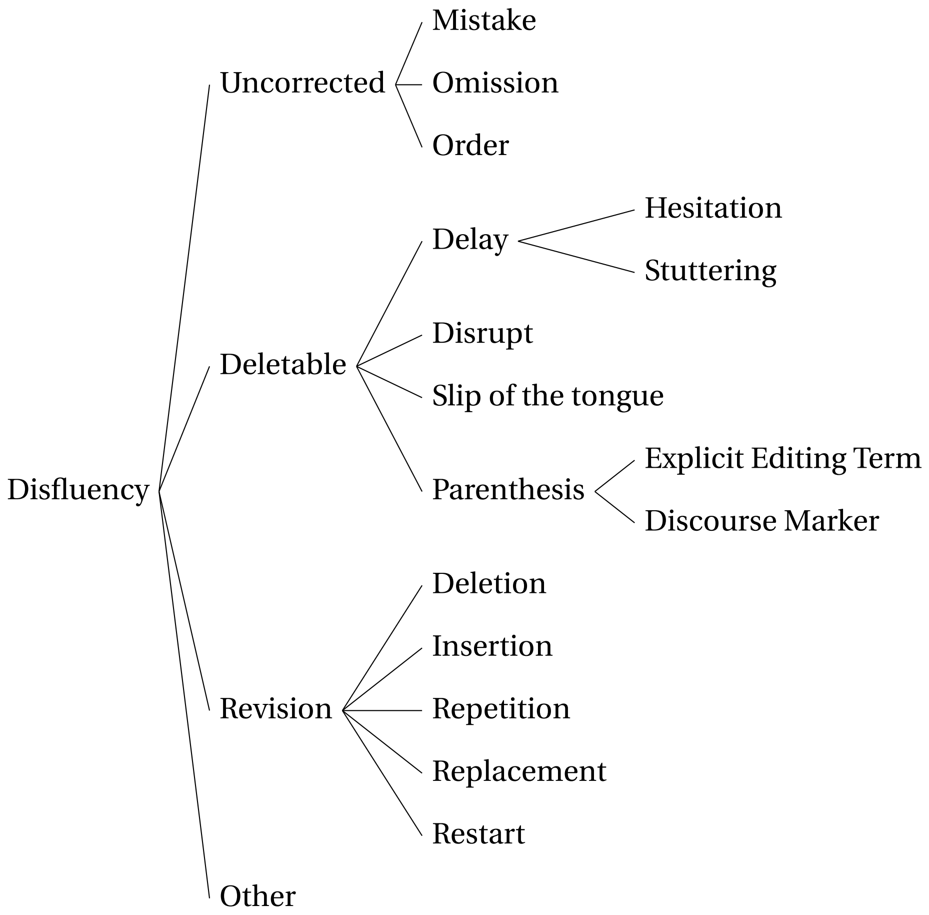


Figure 5.2: Speech disfluency classification scheme for the AMI Corpus (adapted from Besser [2006]).

is too simple, it will not be apt to represent the information required downstream; if it is too complex, an interpretation algorithm will find it difficult to perform the desired analyses. For every feature that adds complexity to the representation formalism we thus have to justify whether that feature can realistically be expected to be achievable by automatic means.

The way a meeting is represented so as to foster subsequent processing is a first substantial difference between the disciplines of extractive and abstractive summarization. The content representation in both cases address different requirements: in extractive summarization, the underlying units or segments, such as sentences or utterances, each are represented in terms of certain features they convey. The purpose of that representation is to support *selection*, namely the selection of the most relevant units (see Figure 5.4). In contrast, the purpose of the content representation in abstractive summarization is to enable *inference* over the content. In-

Original		**Cleaned**	
D:	Well like um,	D:	Well,
D:	maybe what we could use is a sort of like a example of a successful other piece technology is palm palm pilots .	D:	maybe what we could use is a example of a successful other piece technology is palm pilots .
D:	They're gone from being just like little sort of scribble boards to cameras, M_P_ three players, telephones,	D:	They're gone from being just little scribble boards to cameras, M_P_ three players, telephones,
B:	Mm-hmm .	B:	Mm-hmm .
D:	everything, agenda .	D:	everything, agenda .
D:	So, like, I wonder if we might add something new to the to the remote control market,	D:	So, I wonder if we might add something new to the remote control market,

Figure 5.3: Comparison of an excerpt from a manual transcript with and without removing speech disfluencies with GRODI (excerpt from AMI meeting ES2002a).

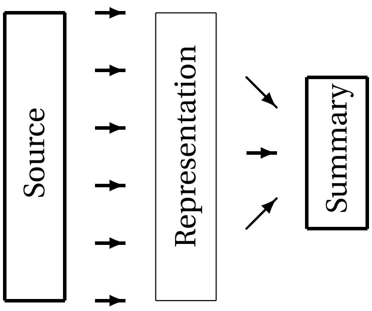


Figure 5.4: Schematic representation of extraction.

ferences are used to abstract the concepts that the representation consists of. That way, content reduction is performed already within the representation, and the final summary text is generated from that reduced representation (see Figure 5.5).

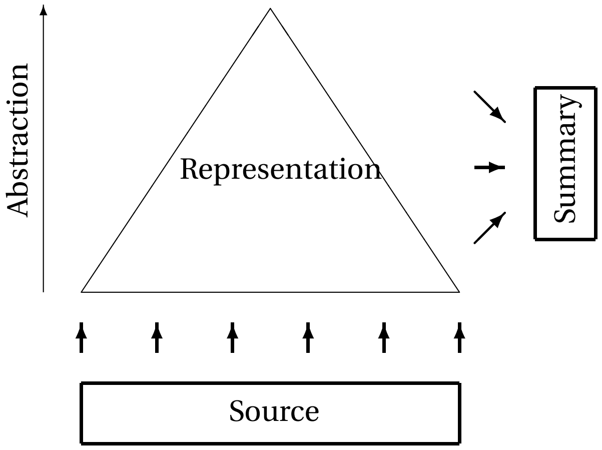


Figure 5.5: Schematic representation of abstraction.

Content Representation

In terms of the general scenario, the work of Alexandersson [2003], in which he summarizes multi-lingual dialogs, is perhaps the approach closest related to abstractive summarization of multi-party conversations. Even though others have worked on meeting summarization (see e.g. [Zechner, 2001] and [Murray, 2008]), these extraction-based approaches are fundamentally different.

Alexandersson's work is conceived in the context of VERBMOBIL, a speech-to-speech translation system [Wahlster, 2000] (see Chapter 3). He leverages the semantic representation formalism of the underlying dialog system. Here, every speaker utterance is represented as a pair consisting of the representation of the utterance's propositional content and the intentional structure of the utterance, encoded as a dialog act (DA).

Representing the intentions of the different utterances is reasonable in the meeting context as well, since it can be paramount for understanding the discourse to grasp the motivation behind a particular utterance and the intended effect a speaker aims to achieve with a discourse contribution. This is not different from the face-to-face dialogs in the case of VERBMOBIL. In fact, an analysis of the abstractive meeting summaries included in the AMI corpus which builds the basis for the research described in this thesis shows that the summary texts frequently contain typical dialog act information. Table 5.1 provides a short illustration with selected examples.

Dialog acts alone are not sufficient though to automatically understand discourse. A dialog act encodes only the intentional content of an utterance, but to summarize a meeting, it is necessary to also have an encoding of the *propositional* content of the utterances.

In VERBMOBIL, Alexandersson uses a knowledge-based approach to model the

Table 5.1: Verbalization of intentional structures in manual summaries in the AMI corpus.

Meeting	Dialog Act	Example sentence
IS1005c	Statement	The project manager stated that the goal for the current meeting was to decide upon a concept for the remote the team is creating.
TS3010c	Suggestion	He suggested using a yellow case with rounded edges and the logo at the bottom, and large, clearly marked buttons.
IS1008a	Request	The Project Manager asked the Industrial Designer to create a functional design plan for the device [...].
IS1000c	Opinion	He likes the idea of implenting [*sic*] speech recognition into a universal remote.

propositions contained in the discourse (see Chapter 3). Others have followed this method, e.g. Castronovo [2009]. Here, the domain of discourse is modeled in form of an ontology (cf. Section 4.4). However, using such an ontology bears a number of disadvantages. On the representation side, the ontology has to be created manually which is not only a tedious, but also a costly process. Because of this fact, ontologies usually restrict themselves to a manageable size. This implies, however, that the author of the ontology is not only an expert of ontology design, but also an expert on the domain in question, and yet in the end the representable content will still be limited to a quite specific domain of discourse. On the interpretation side, since the ontology has to be created anew for every new domain, no all-encompassing theory exists how to instantiate the ontological concepts for a given text, i.e., how to parse a meeting transcript into an ontological representation. Castronovo [2009] and Kleinbauer et al. [2007a] all use a special rule-based parser. But that again requires manual work, namely for writing the rule base, which again calls for expertise in two fields, ontologies and linguistics.

In this thesis, we propose a representation formalism that relaxes the requirements an ontology-centered approach poses. In particular, we aim for a representation that scales up to out-of-domain topics, unlike ontologies, and can offer a more sophisticated way for interpreting the text of a meeting transcript.

The proposed formalism is based on Frame Semantics (see Section) which offers a number of beneficial aspects. First of all, Frame Semantics is a theory of *language*, i.e., it is a natural fit for the interpretation of discourse. This is especially the case through its close relation to cognitive science: the concepts used for Frames are intended to be in concordance with the human cognitive apparatus, which is presumably beneficial not only for automatically understanding language, but also for generating it automatically, because of the close match between surface realization and representation.

Second, with FRAMENET (see Section 4.4), a large-scale implementation of Frame Semantics is readily available. The FRAMENET database is used internationally in different project with different application foci. It is thus well-tested and under constant further development.

Thirdly, methods for deriving Semantic Frames from text have been well studied (see below). Here, another advantage of a large-scale resource such as FRAMENET becomes apparent: with its extensive annotations, it lends itself well to machine-learning approaches to language understanding which promise to scale in accordance with increasing domain sizes as well as showing greater robustness than the more ad-hoc rule-based parsing suggested by e.g. Castronovo [2009].

Another advantage derives from FRAMENET's application agnosticism, enabling usage in diverse scenarios. For instance, a recent research project studies the integration of Frame Semantics early on in a language model for automatic speech recognition[1]. In a scenario like the one of this thesis, which aims at the automatic summarization of conversations, an integration of a single formalism in all stages of language processing promises a more streamlined and thus manageable approach.

However, there are also some aspects that are easier to realize with ontologies as opposed to Semantic Frames. We must not ignore that Frames Semantics is not a theory of knowledge representation, but of linguistic semantics, i.e., the treatment of (physical) entities, their properties and relations is not as well established in Frame Semantics. However, we argue that for the particular task of summarizing meetings, this is not a drawback. We introduce a more shallow treatment of entities below.

In similar fashion to a domain ontology, a model consisting of Semantic Frames first has to be created manually. Because of the more general nature of linguistic Frames as opposed to a special-purpose ontology, such a model is not tied as strongly into a single domain, but more easily re-usable across different domains. In fact, the FRAMENET corpus which we are using as a concrete implementation of Frame Semantics in this thesis, has not been designed with a particular application in mind such as summarization.

A framal representation describes the semantics of a meeting discourse in terms of the situational settings it refers to. The Frame Elements represent, in an abstract sense, the participants in these situations. Frames are evoked by language and likewise the fillers of Frame Elements are textual descriptions. These descriptions may

[1] cf. http://rescue-winbox.calit2.uci.edu:8080/oss_web/oss.htm

encompass further Frames, but ultimately, they reference real-world entities, e.g., the speakers, a presentation, a remote control, a design idea, etc.

Although it is usually fair to assume that it is clear from the context whether we are talking about *actual entities* or whether we are talking about *entity references*, it is important to point out the conceptual difference. Nevertheless, without blurring the distinction, we allow ourselves to refer to both concepts simply as “entities” unless we actually want to highlight that distinction.

From analyzing a meeting discourse presented as a transcript, a summarizer initially encounters real-word entities through references, because that is what is present in the textual representation. The same entity may be referred to multiple times in a discourse, and all references do not have to be the same. A typical example for this effect is the use of pronouns. In the following excerpt from AMI meeting IS1003a, speaker A refers to a certain entity with *your favourite animal* in the first line, and then again to the same entity with *its* in the second line.

[02:52-02:54] **A:** Maybe you can draw your favourite animal
[02:56-03:00] **A:** and make a list of its favourite characteristics.

Co-reference resolution is known to be a difficult problem, cf. e.g. [Mitkov, 2001], but it is clear that in order to produce an abstractive summary, reference handling cannot be ignored. Information concerning a single entity might be spread out over the discourse and thus it is necessary to recognize when such information refers to the same entity in order to put it together and summarize it.

Integrated Content Representation

As noted before, Alexandersson [2003]’s approach employs three separate tiers of representation. A shortcoming of that is that interactions of the represented information requires constant translation between the different formalisms. From the point of view of facilitated downstream processing, a representation that integrated all information sources into a holistic representation formalism would thus be desirable.

This section introduces the concrete representation formalism used in the remainder of this thesis. It consists of three separate tiers of information, too, namely of speaker intentions (dialog acts), propositional content (Semantic Frames) and real-world entities (symbols). We first describe the treatment of propositions with Frames. This includes notes on the representation of referred entities. Finally we demonstrate how the AMI dialog act tag-set can be represented using Frames, allowing for an integrated treatment of propositional and intentional structures.

The reason to represent meeting contents in the first place is because ultimately, we want to be able to generate a summary text. Without such a representation, the only way to generate text would be to manipulate the surface forms we find in a transcript directly. Instead, using a lexicalized representation allows us to prepare the summary contents on a semantic level.

Consider the following sentence, taken from AMI meeting ES2002a:

[01:32-01:36] **B:** So we're designing a new remote control.

According to Frame Semantics, different words in such a sentence each evoke a Semantic Frame, e.g.., *remote control* evokes a Gizmo Frame which contains general words denoting equipment. A Frame analysis of this sentence reveals the following three Frames:

Frame	Invention
Target LU	designing
Frame Elements	• we (Cognizer) • a new remote control (Invention)

Frame	Age
Target LU	new
Frame Elements	• new (Age) • remote control (Entity)

Frame	Gizmo
Target LU	remote control
Frame Elements	• remote control (Gizmo)

The lexical units *so*, *we*, *'re*, and *a* do not themselves evoke Frames. *So* can be interpreted as a discourse marker. In the context of the meeting, the speaker says this particular sentence after a quick introduction round of all meeting participants. We could interpret it as a sign that the speaker intends to begin a new topic, but for a Frame analysis it is of no relevance. *We* refers to the group of meeting participants, i.e. is a reference to real-world entities. The lexical units *'re* (short for *are*) and *a* play syntactical roles, *'re* is a marker of a progressive grammatical aspect while *a* is an indefinite article for *remote control*.

The three Frames encode (at least) three different elements of information:

1. General situation (Frame type)

2. Participants in the situation (referenced entities)

3. Role of the participants with respect to the situation (Frame Elements)

The Frames give us a semantic representation of different pieces of information expressed in a sentence. However, a flat list of Frames does not tell us how these pieces are related to each other, even though such information is desirable for summarization. For instance, the Frame Element INVENTION of the Frame with the same name is *a new remote control.* But this part of the sentence is further described by the other two Frames. This information is not reflected by a content representation built on a set of Frames.

We thus propose an additional structural arrangement of the Frames contained in a sentence that aims at providing additional information on the relation between Frames. When the target of a Frame A is contained inside one of the Frame Elements of a Frame B, we say that Frame A *depends* on Frame B.
More formally, we define this dependency relation as follows:

Definition 5:
The set *Spans* is the subset of $\mathbb{N} \times \mathbb{N}$ so that for every $\{$*(from,to)* $\in$ Spans *from* $<$ *to.* An element of *Spans* is called *span.* For a given span $s \in$ *Spans* we call the projection on its first attribute s_{from}, and the project on its second attribute s_{to}.

A set *Spans*$_{min,max}$ is the subset of *Spans* defined as $\{$*(from,to)*$|$*from* $\geq$ *min* $\wedge$ *to* $\leq$ *max*$\}$.

We also define a containment order over the set of spans.

Definition 6:
The subset $<$ of $(\mathbb{N} \times \mathbb{N}) \times (\mathbb{N} \times \mathbb{N})$ is the set
$< := \{(x, y) | x, y \textit{ are spans} \wedge x_{from} \geq y_{from} \wedge x_{to} \leq y_{to}\}$.

We typically use infix notation for the $<$ relation on spans.

For a given sentence, we can use spans to refer to certain sub-parts of the sentence if we understand the latter as a numbered sequence of characters, including letters, punctuation, whitespace, etc. We can thus think of a Frame analysis of a sentence in terms of an (automatic) annotation of different sub-parts of the sentence. For each annotated Frame Element of the Frame there is a span that references the sub-part of the sentence the Frame Element annotates. Likewise, there is a span that identifies the lexical unit in the sentence which evokes the Frame. We call these *Frame Element span* and *Target span* respectively.

Definition 7:
Let S be a character sequence $c_1, c_2, \ldots, c_n$, and F be a Frame with an associated set of Frame Elements *FE*.
We define a *Frame instance* to be a mapping $FI := FE \cup \{Target\} \rightarrow Spans_{1,n} \cup \{\perp\}$, so that *FI(Target)* identifies the subsequence of S that evokes the Frame, and *FI(fe)* identifies annotated sub-part of the sequence the for each $fe \in FE$. If $FI(fe) = \perp$ for a Frame Element *fe*, then *fe* is not part of the sequence analysis.

Thus a Frame instance is the concrete analysis of a sentence with a Frame. For the sake of brevity, we will still refer to such a concrete annotation as a Frame sometimes, unless the distinction between the two concepts has to be pointed out in the context.

For a set of Frame instances for a given sentence such as in the example above, we can now define a dependency relation with respect to their spans. As discussed above, we want to express that when a Frame's target is contained in one of the Frame Element of another Frame, then the former should be considered dependent on the latter.

Definition 8:
Let $FIs = \{FI_1, \ldots, FI_n\}$ be a set of *Frame instances* FI_i over a sentence. For every i, let FE_i be the set of Frame Elements in the Frame associated with FI_i. We define the relation $depends_{FI} \subset FIs \times FIs$ as follows:
$depends_{FI} := \{(FI_i, FI_j) | \exists fe \in FE_j : FI_i(Target) < FI_j(fe)\}$.

This definition naturally defines a directed graph between the Frames where an edge between two Frame nodes is inserted if they depend on each other. However, since *depends* is a transitive relation, we don't have to insert a node between two dependent Frames f_1 and f_2 e.g. if there is a third Frame f_3 so that f_1 depends on f_3 and f_3 depends on f_1.

Definition 9:
We define a Frame graph as a pair $G = (V, E)$ with $V := FIs$, and $E := \{(f_i, f_j) | depends(f_i, f_j) \wedge \forall f_k : depends(f_i, f_k) \Rightarrow \neg depends(f_k, f_j)\}$.

The graph can by cyclic when there is a sequence of Frame instance $v_1, \ldots, v_k$ with $i \geq 1, v_i \in V$ and $v_1 < v_2 < \cdots < v_k < v_1$. In practice, however, the graphs of actual sentences often tend to be trees, such as the one in Figure 5.6.

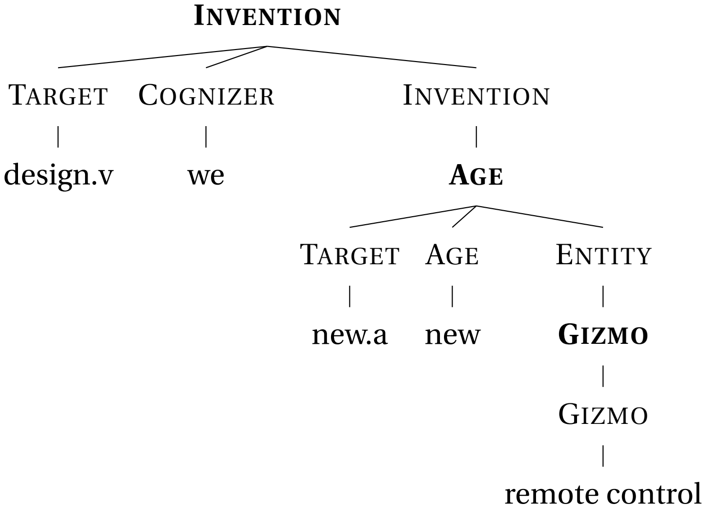


Figure 5.6: The relative position of Target and Frame Element analyses across multiple Frames in the same sentence induce a dependency relation, which can be leveraged to yield a more structured representation of the contents of the sentence.

For summarization, structuring the Frame instances of a sentence analysis in such trees bears some advantages. Dependency means that when a Frame FI_i that is dependent on another Frame FI_j, it describes the semantic meaning of a certain sub-part of one of a FI_j's Frame Elements. This can be seen as FI_j describing a more broad content, while FI_i adding details to what FI_j's Frame Element reflects. As a consequence, the Frame instances closer to the root describe more or less the gist of a sentence while the ones further down in the hierarchy refer to specifics. For instance, in the example the tree makes clear that the original utterance *we're creating a new remote control* is primarily about the *creating* part.

Reference resolution

Some words of a discourse reference entities both from the actual and hypothetical worlds. If we want to be able to follow the discourse of a meeting, it is necessary to be able to identify the items referred to during that discourse. This section is concerned with this particular task which can be divided into three distinct subtasks:

Entity management Bookkeeping about which entities are part of the discourse, where they are first introduced, and which ones are potential candidates for further references.

Reference detection Determining which words actually refer to world entities.

Reference resolution Mapping the the detected references to the entities they refer to.

Ultimately, it's the last of these three tasks that we are interested in for summarization, but the first two tasks are prerequisites for the third. Frame Semantics does not provide a representation of such entities other than their surface forms. Thus, entities become a second tier of information in our content representation.

Assuming a component that can resolve coreferences according to one of the state-of-the-art methods, we collect not only the different occurrences of references for every referent, but also store the reverse direction. This way, the text generation component can choose from the expressions actually used in the meeting to refer to the different entities. Pronouns and deictic repressions should be ignored for this purpose.

Our representation of entities consists to this point mainly of (a) *identity* and (b) *lexical references* and a way to translate between the two. But there is an additional source of information, that the Frame analysis of an utterance provides for entities. When a reference is identical with the Frame Element of an analyzed Frame, and the Frame definition contains a *semantic type* declaration for that Frame Element, we can infer that the referred entity must be of the given type. That kind of information is especially useful for ontological types.

Dialog Acts

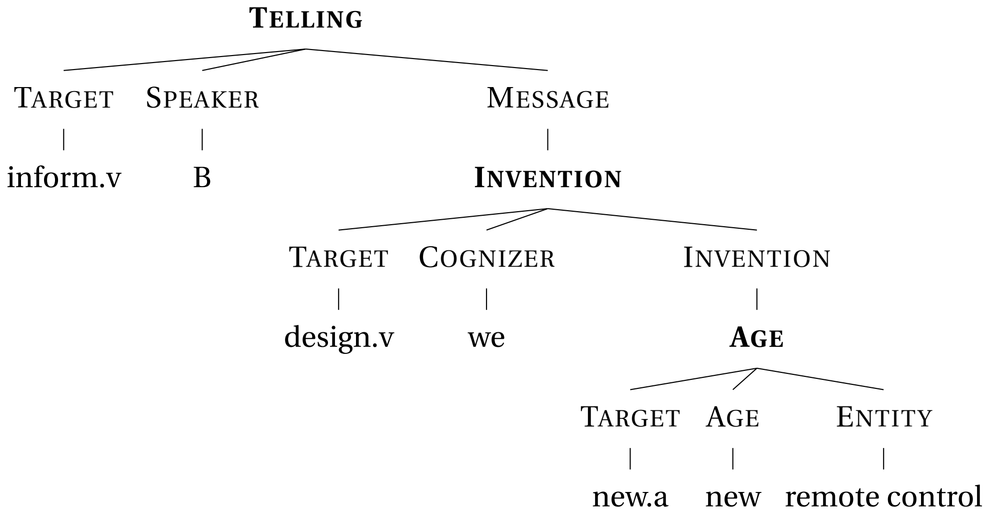
The AMI corpus defines 15 distinct dialog act labels (see Section 4.4), which we are going to integrate into our framal representation. They can be mapped to FRAMENET Frames as shown in Table 5.2.

We do not map the dialog acts *backchannel*, *be-negative*, *be-positive*. The two latter ones occur very rarely (see Figure 4.11), although *backchannels* are quite frequent in meetings. However, they do not add a lot of content and thus are of limited interest for the summarization task. Unlike for other low-frequency DA's there's also no straight-forward Frame candidate to map to among the Frames currently defined by FRAMENET.

The mapping allows us to integrate dialog acts directly in to the dependency trees for Frames, by making the dialog act the root node for all trees contained in an utterance. The above sentence would finally be represented with the following tree:

Table 5.2: Mapping from AMI dialog acts to FRAMENET Frames.

Dialog act	**Frame**
Assess	ASSESSING
Backchannel	-
Be-Negative	-
Be-Positive	-
Comment-About-Understanding	GRASP
Elicit-Assessment	REQUEST
Elicit-Comment-Understanding	REQUEST
Elicit-Inform	REQUEST
Elicit-Offer-Or-Suggestion	REQUEST
Fragment	MAKE_NOISE
Inform	TELLING
Offer	OFFERING
Stall	COMMUNICATION
Suggest	STATEMENT



We now have a complete integrated structure of the propositional and intentional content. More formally, our representation of meeting contents is a pair $rep = (int, ent)$, where $int : DA's \rightarrow Frame\ trees$ is a mapping from every dialog act segment to the dependency tree representation of the Frames contained therein and *ent* is a set of referred entities.

Related work

Discourse interpretation in the most general sense is a vast field of research in its own right, and literature and approaches are manifold. It ranges from very shallow content classification in form of keywords (cf. e.g., [Kleinbauer and Germesin, 2009]) to full-fledged dialog systems (cf. e.g., [Wahlster, 2006], [Zukerman et al., 2008]). Since discourse interpretation is only a sub-topic for abstractive summarization, albeit an important one, and because a extensive report on related work would go beyond the scope of this chapter, we constrain ourselves to the three central points we established above: *dialog act recognition*, *Frame parsing*, and *entity handling*.

Automatic dialog act recognition in a conversational setting is the task of mapping a given stretch of discourse to a set of segments each expressing the speaker's intentionality. Thus it consists of two sub-tasks: segmenting and classification. In earlier works, the focus lay on classification only, i.e., the task of assigning a label from a pre-defined dialog acts to a given segment. The segmentation was done manually. Examples of this task include [Shriberg et al., 1998; Stolcke et al., 2000; Lesch, 2005].

For a realistic scenario, however, the segmentation cannot be taken for granted. Therefore, recent approaches either address the segmentation task separately [op den Akker and Schulz, 2008], or integrate both tasks [Ang et al., 2005; Dielmann and Renals, 2007].

Measuring the quality of a dialog act classifiers is not trivial, as it depends directly on the underlying dialog act tag set. For multi-dimensional tag-sets, such as the e.g. the ICSI MRDA tagset, which allow a very fine-grained and detailed modeling of the speaker's intention, counting exact matches only can be considered too harsh [Lesch et al., 2005]. Smaller tag-sets seem to produce higher scoring classifiers, thus it can be useful to abstract complex tag-sets into a set of top-level distinctions only [Popescu-Belis, 2004].

Another concern for *online* systems that require ongoing classification already as the conversation unfolds, is a small temporal latency. Germesin et al. [2008] suggest an any-time method that provides self-correcting updates of previous classification results when additional information becomes available.

The task of recognizing the Frame semantic representation of a given piece of text can be divided into two sub-task that are sometimes approached separately: *Frame target prediction* and *semantic role labeling* [Gildea and Jurafsky, 2002].

The first task is that of identifying lexical units in the source text that evoke one of a set of known Frames, and determine which Frame that is. The second task assumes that the evoked Frame is known, and calls for the prediction of the Frame Elements of that Frame (cf. [Màrquez et al., 2008]).

Despite growing interest in these tasks, the number of available systems is rather small. In addition, some semantic role labeling system are based on other semantics theories than Frame Semantics, e.g. [Pradhan et al., 2004; Matsuzaki and Tsujii, 2008]. Although it is conceivable that such a system could be employed with the help of a mapping of the employed theory to Frame Semantics, parsers that predict Frame Semantics natively are preferred.

The SHALMANESER toolchain [Erk and Padó, 2006] is a flexible machine learning architecture for Frame analysis of raw text. It uses lexical features such as bag-of-words, bi- and trigrams around the target word, voice of verbs, etc. The processing pipeline consists of three steps: pre-processing of the input text, including syntactic parsing and lemmatization of lexical units, Frame target prediction, and–based on the output of the previous step–Frame Element prediction. SHALMANESER provides pre-trained models for English, based on the FRAMENET corpus, and German, based on the Salsa/Tiger corpus [Burchardt et al., 2006]. SHALMANESER reaches an accuracy score for Frame target prediction of 0.932 for English and 0.79 for German. It predicts Frame Elements with an f-scoreof 0.75 for English and 0.6 for German.

Das et al. [2010] present a similar system, called SEMAFOR. Their pre-processing step adds dependency parsing of the input text, similar to the method of Johansson and Nugues [2007]. The latter method is used as a baseline for the SEMAFOR evaluation, as it was the best performing approach in the SemEval'07 shared task.[2] In contrast to SHALMANESER, the target identification phase in SEMAFOR does not use machine learning, because the authors found a tendency for overfitting to the training data. Instead, they use a small set of rules to identify Frame evoking lexical units. Machine learning is used, however, to train a model for predicting the evoked Frame, and for the third phase, the prediction of Frame Elements. The resulting system outperforms the baseline at every stage of processing. A new release, version 2.0, of SEMAFOR has been published in 2011.

Finally, we take a look at previous work in identifying real-word entities in texts. Coreference resolution is closely related to a well-established field of natural language processing, *anaphora resolution*. It differs from that task, however, in that anaphora resolution takes the referents from the linguistic material of the discourse. Anaphors are resolved by mapping them to antecedents. In contrast, we are also interested in the real-world entities a reference refers to.

[2] cf. `http://framenet.icsi.berkeley.edu/semeval/FSSE.html`

5.5 Transformation into Summary Contents

The main goal of summarization is to create a compacter document than the original source that still contains all or most of the relevant information. The objectives of the *transformation* step which we describe in this chapter are thus

- to determine what is relevant and
- to compact the information deemed relevant

Macro-rules and Frames

For our own approach, we revisit van Dijk's and Kintsch's macro-rules as described in Section 3.2:

Deletion : Given a sequence of propositions, delete each proposition that is not an interpretation condition (e.g., a presupposition) for another proposition in the sequence.

Abstraction : Given a sequence of propositions, substitute the sequence by a proposition that is entailed by each of the propositions of the sequence.

Construction : Given a sequence of propositions, replace it by a proposition that is entailed by the joint set of propositions of the sequence.

Figure 5.7 illustrates the three macro-rules in a schematic way. Here, a source document is represented in terms of the propositions it contains, symbolized by the different circles. The *deletion* rule simply discards some of these propositions, as alluded to by the purple crosses. Arrows symbolize logical entailment; *abstraction* occurs when a proposition is entailed by one or more separate propositions. The *construction* rule applies when a proposition is entailed by a set of propositions. In both cases, the summarizer may choose to include only the entailed proposition in the summary. The difference between abstraction and construction is that in the case of abstraction, each of the identified propositions entails the target proposition, while in case of construction, it is the set of the identified propositions as a whole.

We also note that some of the macro-rules may already have been applied by the author of the source, so that the resulting propositions can be found in the source itself. It is more typical, however, that the summarizer has to apply the macro-rules, in which case the resulting propositions are not part of the source. The final summary is thus likely to consist of both inherent and derived propositions.

The classic representation formalism for the propositions found in text is first order logic. In our approach, however, we have described before how we leverage

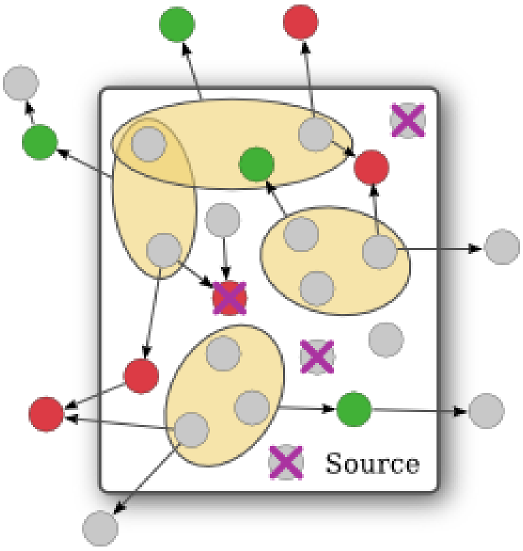


Figure 5.7: Illustration of the macro-rules *deletion*, *abstraction*, and *construction*. Abstraction and construction derive new propositions from existing ones, while deletions remove propositions.

Frame Semantics instead. It is thus necessary to adapt the notions used in the original description of macro-rules to Frames.

By definition, both *abstraction* and *construction* rely on a notion of logical entailment based on the propositions of the source text. This task – textual entailment – is a difficult problem in general. In fact, there is a whole sub-field of linguistic research dedicated to it (cf. e.g. [Androutsopoulos and Malakasiotis, 2010]).

Deletion

The deletion of propositions, or more general, of parts of the source can be seen as the most basic operation in summarization. In fact it is arguably the only rule used by extractive summarization approaches. *Deletion* is more basic than *abstraction* and *construction* because these two rules make themselves use of a deletion operation: they delete certain propositions and introduce an entailed proposition instead. The *deletion* macro-rule itself, however, does not require a replacement the way the other two rules do, it simply removes source material. In combination with *abstraction* and *construction*, it is reasonable to apply *deletion* as the final step, only after the other two rules can no longer fire. If we applied *deletion* first, we would run the risk of deleting material which in itself may not be interesting, but which could contribute to an *abstraction* or in particular a *construction*.

Since it does not offer a replacement for the deleted parts, this rule can be seen as a more drastic or unforgiving operation, and thus its application should be justified in each case. This is especially true when replied in mutual recursion with the other rules. They successively replace content with more concise (but less detailed) content. After a number of recursive steps this means that a larger number of original propositions have been replaced by a small number of entailed (macro-) propositions. Applying the *deletion* rule to one of these proposition then comes down to deleting all of the original propositions that contributed to the application of the other rules.

As [van Dijk, 1980] discusses, the *deletion* rule can be seen from two opposite views: a summarizer can either keep all propositions by default and *delete* the irrelevant ones, or they can discard all propositions by default and only *select* the relevant ones for inclusion in the summary.

In a computational context, the decision which parts of a given source to delete and which parts to select needs to be informed by some observable properties of the source. Here, we do not restrict "source" to mean only the transcript, in fact, non-lexical features in speech-related summarization have been shown to contain valuable information. Since feature sets have been extensively researched in extractive approaches, we can draw from state-of-the art results for meeting-related extraction (cf. [Kleinbauer and Murray, 2012]):

Prosodic features Prosodic features (sometimes called *suprasegmental* features) are low-level features extracted from the voice signals of the participants. They include the pitch contour, signal energy, junctures, and rate-of-speech information.

Such information is useful for detecting high intensity phases in the meetings ("hot spots") which may be an indicator for particularly interesting parts. Prosodic features also build the base for the extraction of some of the other features mentioned below. For instance, the pitch contour may be indicative of question dialog acts, and longer pauses could be used as hints for structural analyses.

Visual features Non-verbal, or more particular, non-audio features can offer information that cannot be found in a speech transcript. For instance, agreement to or rejection of a proposal could be indicated by some of the participants through nodding or shaking the head Zobl et al. [2003]; Yang et al. [2002]. Likewise, gestures and movement in the meeting room could provide further relevant information.

In order to access this information, it is required that the meeting room provides camera to capture visual cues. Typical features extracted from a video stream include Global-Motion [Zobl et al., 2003] or skinblobs [Yang et al., 2002] for recognizing the location of hands and faces [Arsić et al., 2007]. Such low-level features may first pass through additional intermediate interpretation to yield gesture or facial expression information.

But it is not only the behavior of the meeting participants that can be analyzed visually. Other relevant cues can be provided by, e.g., tracking changes in slide presentations, or real-world objects (such as design prototypes) in the meeting room.

Lexical Features Features extracted from the words in the transcript of a meeting are called *lexical features*. The quality of these features in a fully automatic setting depends largely on the word error rate of the employed ASR system. For extractive summarization, however, some experimental results suggest that extract-worthy segments correlate with high recognition rates per segment, so that more severe ASR errors are somewhat compensated [Valenza et al., 1999; Zechner and Waibel, 2000; Murray et al., 2005].

Standard lexical features include term weighting schemes such as e.g., *tf.idf* or *su.idf* [Murray and Renals, 2007], n-gram statistics, i.e. frequencies of sequences of n words as they appear in the transcript, and the occurrence of certain cue words or phrases.

Structural Features While structural features have been found quite indicative of the extract-worthiness of sentences in document summarization, meetings often exhibit less structure due to their spontaneous component. Nevertheless, some features such as the whether a textual unit appears in the beginning or toward the end of a meeting, where the participants wrap-up and recapitulate the meeting, can be beneficial. Other structural features dialog acts, and adjacency pairs of dialog acts (e.g., question-answer pairs).

A higher level structural feature is the topical organization of a meeting (see also Chapter 4).

A Priori Knowledge Some behavioral features of the meeting participants such as a certain vernacular or turn-taking behavior, that could be beneficial for extracting utterances, may have lead to skewed results unless these effect get normalized by such features gender, origin, social and professional status, of the different participants.

Approaches to detect some of these features automatically have been proposed (e.g. [Müller, 2006]), but in general it is fair to assume that such information about speakers can be collected beforehand and could be made available to an automatic summarizer.

Extractive approaches operate on textual units, such as sentences or dialog act segments, and not on propositions. A set of features alone, however, does not yet provide an extraction mechanism. For that it is crucial how to combine the different features and how to compute a *significance* function f that yields for every unit of text either 1 (keep) or 0 (delete). Most modern approaches employ machine learning techniques to implement such a significance function.

Since we are ultimately interested not in the flat surface representation we get from the transcript, but in the richer representation output by the interpretation phase, we have to adapt the extractive deletion mechanisms to work with our version of propositions. Under the assumption that the interpretation phase of Chapter 5.4 delivers propositions in some formalism for a given segment of a meeting transcript, and that there exists a feature analysis component in the manner of extractive summarizers that can evaluate the same segment of the transcript, we suggest the following rule for *deletion*:

> *Let P be the set of propositions contained in a segment S of the meeting transcript, and d an extractive significance function. Then delete a proposition $p \in P$ if and only if:*
>
> 1. *p is not used by any* abstraction *or* construction *rule*
>
> 2. $d(p) = 0$

This rule lifts text-based deletion to propositions extracted from that text. A variation would be to only allow those propositions to contribute to *abstractions* or *constructions* that do not stem from text marked as deletable by the given significance function. With the version presented above, though, we implicitly implement the thought that *deletion* rules should be applied as the last of the macro-rules.

In the most simple case, *deletion* is only applied to propositions extracted immediately from the source. However, as the illustration of Figure 5.7 suggests, it may also be applied to the output of the other two rules. Such a case is not covered by the lifting rule above and would require additional conditions under which *deletion* could fire.

Abstraction

The *abstraction* rule applies when one or more propositions found in the source entail a *target* proposition. In that case, all of the source propositions are replaced by the target proposition. If the target proposition is itself part of the source, then the *abstraction* rule is indistinguishable from the *deletion* rule applied to the source propositions. If the target proposition is not contained in the set of all source propositions, we typically expect the number of source propositions that trigger the *abstraction* rule to be at least two. Otherwise we effectively replace one proposition by another one, which does not bring us much closer to the goal of reducing content.[3]

[3] Of course there are cases, where content reduction can be achieved even when only a single source proposition implies the target, for instance, *the participants introduced themselves by name and role in the project* entails *the participants introduced themselves* which certainly constitutes a content reduction.

Naturally, the number of propositions entailed by any given proposition can be quite large, and at the same time the implied propositions are often times completely irrelevant for summarization. For instance, consider a meeting utterance like the following:

B: We used to live in London when I was a kid.

From that we can deduce among other things that the speaker is no longer a child. But that information is hardly worth including in the summary of the meeting:

* Speaker B is an adult.

Such a proposition should be subject to a later *deletion* rule. But it would be more economic not even to infer unnecessary propositions that are bounds to be deleted again. Also, such an inferred proposition may still be useful as an intermediate proposition that helps trigger a second *abstraction* or a *construction* rule.

In the following, we discuss two specific kinds of deriving entailed propositions that actually reduce information (instead of adding unnecessary information).

Frame Element Reduction

A potentially better way to transform the content of the above sentence into a summary (assuming it should not be deleted altogether) could be:

Speaker B used to live in London.

This sentence is certainly entailed by the original utterance. In comparison to B's original utterance, two major content-wise changes can be observed: the change from *we* (presumably speaker B's family) to *Speaker B* alone, and the deletion of *when I was a kid.* We concentrate on the latter effect.

Let us consider a Frame analysis of the main proposition in B's utterance (see Figure 5.8). It is straight-forward to see that the proposed summary sentence has a similar analysis, except that the Frame Element TIME is missing. This inspires the first entailment rule we consider for *abstraction*: leaving out Frame Elements.

The choice of Frame Elements to leave out in order to reduce content can obviously not be arbitrary. For instance RESIDENCE or LOCATION could not be left out:

- *Used to live in London when he was a kid.
- *Speaker B used to live when he was a kid.

Frame	Residence
Target LU	live.v
Frame Elements	• We (Resident) • in London (Location) • when I was a kid (Time)

Figure 5.8: Frame analysis of a sample utterance

These two FE's are conceptually necessary for the Residence Frame which is expressed via the *Coreness* status of Frame Elements in FrameNet.

Note that this is merely a heuristic. Depending on the focus of interest, Frame Elements such as Time may in fact be very relevant, for instance, when considering a meeting of historians.

Frame Generalization

A second kind of logical entailment is given by the hierarchy of Frames as laid out in the FrameNet corpus. By definition, every instance I of a Frame F is also an instance of the super-frames of F. This allows for three kinds of content reduction.

The first kind of content reduction happens when a Frame Element of F that is instantiated in I does not exist in the super-Frame of F. In that case, lifting I to F's super-Frame consequently means that said Frame Element has to be removed.

A similar case can occur when a Frame Element of F does exist in the super-Frame, but while it has *core* status in F, the FE in F's super-Frame does not have *core* status. In that case, the FE could be dropped again using Frame Element reduction as described in the previous section.

When lifting a Frame to one of its super-Frames, and especially when dropping Frame Elements in doing so, we have to ensure that the target LU is also in the super-Frame, or else, we have to exchange it appropriately for an LU that is.

A third strategy for content reduction using the Frame hierarchy is to generalize multiple instances of different Frames to the same super-Frame. The main advantage of such a step is to allow to aggregate the Frame Elements of the instances of the (now equal) Frames in a second step and thus express their content in a single, aggregated Frame. Aggregation can either be performed as a generator device, i.e., on a linguistic level, or on a representation level by grouping and ungrouping similar entities (see Section 5.5 below).

As an example, consider a meeting in which two events happen that can be represented by the following Frames[4]:

Frame	RECAP	INTRODUCTION
Target	recap.v	introduce.v
Speaker	the project manager	the project manager
Topic	the results of the previous meeting	the agenda of the current meeting

Without applying the *abstraction* rule, these two Frames could be generated in a summary as separate sentence:

> *The project manager recapped the results of the previous meeting.*
> *The project manager introduced the agenda of the current meeting.*

But we notice that these sentence express very similar content, and in fact, assuming that both Frames inherit from the more general STATEMENT Frame, we could apply the *abstraction* rule to lift the representations:

Frame	STATEMENT	STATEMENT
Target	talk.v	talk.v
Speaker	the project manager	the project manager
Topic	the results of the previous meeting	the agenda of the current meeting

Note that this lifting involved changing the target LU's so that it evokes the more general STATEMENT Frame in both cases. The SPEAKER FE is identical now for both sentences, and the TOPIC FE's can be grouped, allowing to aggregate the two sentences above into a single sentence:

> *The project manager talked about the results of the previous meeting*
> *and the agenda of the current meeting.*

Effectively, we have made the statement less precise, because we replaced the more informative *recapped* and *introduced* with the more general *talked.* At the same time though, we have managed to shorten the summary by a few words.

When we assume a hierarchy with a single root Frame, it is always possible for any two arbitrary Frames to be abstracted to a common ancestor. This naturally

[4] These Frames are not part of FRAMENET, but introduced here for illustration.

poses the question whether there should be a limit to how far the *abstraction* rule can abstract propositions away. Otherwise we risk losing to much information on the path from the source propositions to the ancestor. Summarizing a set of propositions expressing such as:

> *The project manager opened the meeting and introduced the agenda of the current meeting. The marketing expert gave a presentation on the latest consumer trends. The industrial designer presented the current prototype, while the interface designer prepared a critical review of existing remote controls.*

to an over-generalizing INTENTIALLY_ACT Frame such as:

> *Everybody did something.*

is utterly useless. We thus suggest using heuristic limits which we motivate as follows.

Limit by path length When looking for a common ancestor Frame for a number of given candidate Frames, one constraint can be to limit the number of nodes between the common ancestor and each candidate. This is a straight-forward heuristic that attempts to estimate the level of abstraction by the number of Frames on the path between a candidate Frame and the result Frame of applying the *abstraction* rule.

This heuristic is quite conservative, as it is guaranteed to always be applicable. It is able to limit the degree of abstraction allowed, independent of the candidate Frames. On the downside, it is arguable whether the number of levels in a hierarchy is a good measure for the degree of abstraction, because a more finely worked out part of the Frame hierarchy may contain more levels than a more coarsely crafted part. But that does not necessarily mean that the conceptual abstraction is higher in the first part.

Only lexical Frames on paths Non-lexical Frames in FRAMENET are Frames that cannot be evoked by any lexical units, but exist solely as conceptual parents for further, more specialized Frames. For that reason, non-lexical Frames cannot be used as the outcome of an *abstraction* rule, because without any evoking LU's, they could not be verbalized in the summary text.

A natural way to overcome this heuristic is to move to ancestor of the non-lexical Frame that is a lexical Frame. That would, however, have the reverse effect from what this heuristic tries to achieve. Instead of limiting the level of abstraction, using a lexical ancestor implies using an even more abstract Frame as the output of the *abstraction* rule in question. It seems more reasonable, to require that an *abstraction* rule may only fire if all Frames on the paths between candidate Frames and result Frame are lexical Frames.

Do not pass Frame with core-unexpressed FE's When a Frame Element of a given Frame is marked as core-unexpressed, that means that sub-Frames may not inherit that Frame Element. Conceptually, the sub-Frame still expresses the information described by that Frame Element, but this may happen implicitly, for instance, in the lexical units that evoke the sub-Frame (cf. Section 4.4).

In turn, when moving up the hierarchy to find a common ancestor for a set of candidate Frames, an *abstraction* rule might encounter an ancestor Frame that has a core-unexpressed Frame Element. In that case, it is not easily achievable to automatically transport to that Frame the implicitly encoded information of the child Frame. Therefore, this third heuristic advocates limiting the target Frames of the outcome of *abstraction* rules to Frames that do not define core-unexpressed Frame Elements.

Construction

The third macro-rule we consider is the *construction* of a new Frame, when it is logically entailed by a set of Frames as a whole, i.e., the new Frame is not entailed by every single Frame from the set, but only from the interplay of all involved Frames.

One way this rule can be interpreted is to infer a certain higher level Frame when all (or some) the Frames it *uses* (cf. Section 4.4) are recognized. This approach is reminiscent of the notion of *sketchy scripts* as used in the FRUMP system (see Section 3.4). For instance, let us assume the existing of a framal description of having a meal at a restaurant. If we observed in the source representation Frames for entering a restaurant, ordering food, eating the food, and paying the meal and leaving, a *construction* rule could replace these four Frames with a single instance of our RESTAURANT Frame (cf. 4.3). Such a content reduction is valid because the process of going to a restaurant is conventionalized in a way, that a reader of a summary could deduct the original four propositions from their own world knowledge and the constructed Frame.

In order for apply *construction* rules in the way just outlined, we thus require script knowledge, i.e., knowledge about the decomposition of complex framal situations into the sub-Frames they are made of. The FRAMENET corpus provides such information in form of the *uses* relation.

However, additional scripts may be necessary to cover a wider range of construction possibilities. The current implementation of the MEESU system contains a knowledge base of manually created scripts. They were derived by inspection of the annotations available in the AMI corpus: for every sentence in a meeting summary, the corpus provides reference links to utterances in the meeting transcript that support the content of that summary sentence. For instance, when the summary sentence refers to a certain discussion during the meeting, the references link to the portion of the meeting in which the discussion took place (see also Section 4.4).

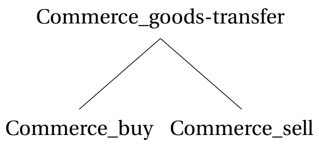
This structure can be leveraged to construct script information. In future work, machine learning could be a promising approach to construct such information automatically. At the moment, the scripts have been constructed by judging the Frame content of the summary sentence and the linked utterances. For instance, “giving a presentation”, an activity which can be found frequently in the AMI summaries, is often manifested in a longer sequence of TELLING frames in the analysis of a transcript. Therefore, a script has been formalized that constructs a CAUSE_TO_PERCEIVE frame with *present.v* as the target when a sufficient number of TELLING Frames with the same speaker are found in sequence in a meeting representation.

Information-retaining Transformations

In addition to the macro-rules described above, we also consider two further transformations that do not themselves reduce content, but transform information from one form of representation into an equivalent, or reversible, second form. Although such operations keep the information content constant, they can nevertheless be useful for summarization, because the representation as produced by the outcome of the application of such a rule may enable one of the three macro-rules, while the original representation does not. In that way, they indirectly contribute to content reduction.

Change of Perspective

FRAMENET encodes different perspectives on an event through the *perspective_on* relation. Section 4.4 illustrates this with the Frame COMMERCE_GOODS-TRANSFER which can be seen from the perspective of the buyer (COMMERCE_BUY) or from that of the seller (COMMERCE_SELL). Both perspectives encode the same situation though.



Therefore a change of perspective does not add or lose any information, it may however simplify the integration of the encoded information into a compact representation. By change of perspective we mean that given a Frame, e.g. COMMERCE_BUY, it is possible to replace it with the non-perspectivalized Frame COMMERCE_GOODS-TRANSFER or even with the opposite perspective COMMERCE_SELL, as both are logically entailed.

An example, why a change of perspective can be useful, is given by the following sentence:

> *Alice bought a bike from Bob last year, and this year he sold her his car, too.*

The first part of the sentence evokes COMMERCE_BUY while the second part evokes COMMERCE_SELL. By changing perspective on the first Frame, we can represent both events with the same Frame, COMMERCE_SELL, which allows us to simplify the sentence structure as follows.

> *Bob sold his bike and car to Alice.*

Using the *abstraction* macro-rule twice, we have dropped the temporal information, and aggregated the goods to yield a short summary sentence conveying essentially the same information. Similarly, a script-like description consisting of multiple scenes may preferable use the neutral COMMERCE_GOODS-TRANSFER in one of the sub-scenes, as the perspective is of secondary order. Thus being able to infer the non-perspectivalized Frame from its perspective can enable the *construction* macro-rule that would otherwise not be triggered.

Grouping of Entities

This transformation is concerned with the representation of real-world entities, and in particular the relation between groups and the members of groups. As far as our representation is concerned, a group is a *reification* of the members it contains, i.e., it allows to refer to the set of member elements *as a whole* with a single referent.

This is indeed an important concept that is closely related to aggregation in NLG, except that it refers not to linguistic surface forms, but to the Semantic representation.

The basic idea is two-fold. For one, we would sometimes like to be able to fill a Frame Element with multiple entities when that is opportune, because it allows us to make a statement involving multiple entities with a single proposition, naturally compacting the resulting summary text. In turn, if a proposition involves a group of entities, but only one of them is considered relevant for summarization, it is desirable to have mechanism to “break up” the group representation and extract the single entity. Again, this will result in a compacter summary because we can discard unnecessary information.

For meetings, a straight-forward example of this is the distinction between the meeting participants as a team or group versus the representation of each participant as an individual. Consider for example, a representation of the following three propositions:

- The marketing expert gave a presentation.

- The user interface designer gave a presentation.
- The industrial designer gave a presentation.

Aggregation of surface forms would allow a generate to collapse these separate propositions into a single sentence like this:

> *The marketing expert, the user interface designer, and the industrial designer gave presentations.*

This is a bit clunky. What is more, it is in so far ambiguous as we have lost the information that every one of the involved people gave a presentation of their own. The above sentence also allows a reading in which the three people as a group gave multiple presentations together. This ambiguity could at least be fixed by simple variations of the sentence:

> *The marketing expert, the user interface designer, and the industrial designer each gave a presentation.*
>
> *The marketing expert, the user interface designer, and the industrial designer gave presentations to each other.*

However, aggregating that way has it its limitations. Imagine the meeting group did not consist of three or four people, but of ten, and each of them gave a little presentation in a meeting. Certainly, we would want to avoid the generation of a sentence in the above manner, where every single participant is mentioned explicitly. By reifying the participants as a single group, an automatic summarizer could instead produce a more natural sounding sentence such as:

> *The group gave presentations to each other.*

Note that this kind of aggregation cannot simply happen on a surface form level, because it presupposes a conceptual understanding about when it is appropriate to replace a set of individuals with a reification thereof. For an automatic treatment of entity grouping/ungrouping, the first question that arises is thus under which conditions it is valid to perform a reification. Valid here means first and foremost that a reader of the resulting text can be expected to be able to reliably identify the reified entities. For instance, using a definite reference such as *the group* in the above example implies every participant in the meeting; if a proposition is true for only three out of four people, it would be misleading to refer to them as *the group*.

On the other hand, it is fine to use a reference to the whole group even though only three of the four people are meant in a sentence such as:

> *The project manager introduced the upcoming project to the team.*

The project manager is part of the team, yet it is hard to conceive that he first learns about the project through his own introduction. Here, *the team* refers to everybody in the team except for the project manager himself. Such fine-grained distinctions present a challenge for a generation system.

A second question is whether a proposition using a reified group entity is to be understood as referring to the group as a whole or to the members of the group individually. The problem arises from the fact that a reified entity is itself an entity, and at the same time is a place holder for other entities. Thus the proposition containing such an entity could refer to the reifying entity, or to the reified entities. The following two sentences illustrate the distinction.

1. The participants introduced themselves to each other.

2. The participants decided to meet again later that day.

In the first sentence, every single participants performed an introduction, thus assuming the standard group size of four participants, four distinct introduction events occurred. But in the second sentence, not everybody of the group made their own decision to meet again later - rather, this sentence should be read expressing that the decision was made collaboratively, i.e., only one decision event occurred.

Having a way to represents groups even allows a summarizer to use quasi-mathematical set operations, e.g. set difference, to generate a sentence such as:

> *Everybody except the project manager gave a presentation.*

5.6 Text Generation

Introduction

The work laid out in the previous chapters creates a representation of the most relevant parts of a meeting and could thus be considered a summary of a meeting. However, the form of representation is not very accessible for people and in fact it is fair to expect a summary to be presented as a coherent piece of text.

The process of producing a textual serialization of symbolic representation structures is referred to as *Natural Language Generation* (NLG). We refer to a component that produces English text from a summary representation as an *NGL system*, or short, a *generator*. An NLG system takes as input a content representation and generates text that verbalizes the information of its input.

Traditionally, the problem of generating natural language texts can be seen from two complementary points of view, namely *what to say* and *how to say* it, i.e. content determination and content verbalization. In Reiter and Dale [2000]'s reference architecture for NLG systems, shown in Figure 5.9, the "what to say" is realized by the

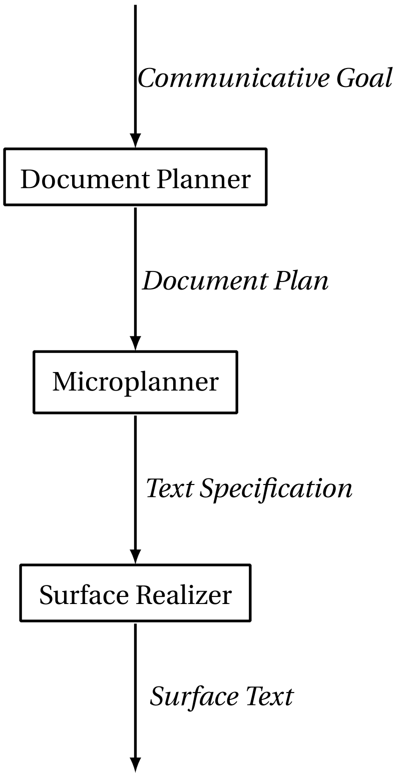


Figure 5.9: Reiter and Dale's standard architecture of a generator pipeline [Reiter and Dale, 2000, p60].

Document Planner component, while the "how to say" is determined by the Document Planner and two additional components, *Microplanner* and *Surface Realizer*. Reiter and Dale enumerate six different tasks that the three components accomplish.

Content Determination The task of the Document Planner is to decide which information from the given input should be included in the generated text. In our case, this is mostly determined by the *Transformation* component of the summarizer (see. Chapter 5.5), but in the general case, the generator may have to select the content from a database of information (e.g., meteorologic data used for an automatic weather report system [Goldberg et al., 1994]).

Document Structuring This is also a task of the Document Planner. The goal is to create a plan of the full text to generate, i.e., composing the information content into concrete logical units. These units are related to each other through rhetorical relations. As the name of the task suggests, the result of this task is a structural view of the document.

Lexicalization This task is responsible for selecting the suitable linguistic material with which to verbalize the content.

Referring Expression Generation This task is closely related to lexicalization, and both task belong to the Microplanner component. It is concerned with the mapping of real-world entities to linguistic expressions, i.e., find words that uniquely identify each entity in question.

Aggregation In the aggregation phase, the Microplanner determines the concrete linguistic structures,e.g., paragraphs and sentence, for the units planned by the Document Planner. Within the given limits, the aggregation phase may also decide on structural aspect such as the order in which information in a logical unit should be expressed.

Linguistic Realization Building upon the result of the previous task, linguistic realization puts everything together to produce the actual, continuous text.

Structure Realization The last task addresses technical issues of text markup or file formats. It makes sure that structural information determined before, such as, paragraph information, is kept intact in the output format.

In this chapter we first give an overview of the principles of NLG and then introduce a novel approach that is tailored toward a Frame-based representation. We demonstrate how the rich annotation found in the FRAMENET corpus can be leveraged to create partial syntactic trees which we use to generate English sentences.

Related Work

The literature on the generation part of abstractive summarization is rather scarce. The reason for that is that most authors concentrate on the *interpretation* and *transformation* phase in their work. Natural language generation is perhaps seen as an orthogonal field of research because it has many applications outside summarization. However, we argue that the final text generation is an important step that cannot be left out, because the whole purpose of a summarizer is to produce a written summary. In particular, it is difficult to make any qualitative or quantitative statements about summarizer as a whole, if there is no final summary that can be evaluated.

In particular, research on text generation particularly for summarization is rare. A notable exception though is McKeown et al. [1995] who describe two systems that generate summaries of basketball games and planning activities respectively. For the use of more general NLG systems in abstractive summarization, we refer the reader to our detailed descriptions in chapter 3.

Text generation approaches leveraging the FRAMENET database are equally rare, even outside the field of abstractive summarization. De Bleecker [2005] proposes an approach for a generator that uses FRAMENET to determine lexical choices within a dialog system. However, her proposal does not contain any details how to proceed with the generation of a Frame, once the lexicalization has been determined. Besides, she does not have seem to actually have implemented her approach in an actual system.

Thus to the best of our knowledge, the approach described in the following sections is the first implementation of a generator that is able to generate English sentences from a Framal input representation.

Approach

As in previous chapters, we understand the summary and its representation to realize different abstract situations, e.g., discussions, presentations, etc. For every sentence of the summary, the generator receives as input a description of the situation (or even multiple situations at once) in some sort of representation formalism. In general, this could be raw data, logical forms, ontologies, or others.

We propose here to use Semantic Frames to describe the different situations that are to be verbalized by the system. This is a straight-forward idea, since Frames are especially conceived to represent situations. However, they have mainly been used for *analyzing* text in the past, not for *generating* text.

According to the theory of Frame Semantics, Frames are evoked by certain lexical units. In turn, that means that in order to communicate the situation described by a certain Frame, the generated sentence should contain one of the lexical units that evoke that Frame.

The output of the previous summarization phases not contain Frames alone, but pairing of Frames and lexical units. This is because a Frame alone does not contain information how the situation represented by it can be verbalized as a correct sentence. For instance, passing an EXPERIENCER_FOCUS Frame to the generator without a lexical unit does not contain sufficient information how to verbalize that Frame as both of the verbs *love.v* and *hate.v* evoke EXPERIENCER_FOCUS, as do a number of other lexical units.

The first choice a generator has to make thus involves selecting the desired Frame-evoking LU. Since the generator input consists of pairings of Frames and LU's, a first choice for the target LU is readily available. However, if that first choice is incompatible with the subsequent steps, the sentence generation may only succeed when using an alternative target lexical unit (see Figure 5.10).

Given a Frame and an evoking lexical unit, there are potentially many different variations of how the Frame can be realized syntactically. This depends mainly on the valences of the lexical unit and which Frame Elements are actually present.

But also stylistic considerations may influence the choice. Consider the following example sentences, for a hypothetical CAUSE_TO_PERCEIVE Frame. In the first two sentences, a single Frame Element AGENT is realized as “the project manager” while it is absent in the third variation.

1. *The designer gave a presentation.*

2. *A presentation was given by the designer.*

3. *A presentation was given.*

4. **Gave a presentation the designer.*

In the first three sentences the noun *presentation* together with its support verb *give* is the lexical unit that evokes CAUSE_TO_PERCEIVE. The first two sentences are synonymous to each other, but perhaps the active voice of the first variation is preferred in terms of writing style. The third alternative, however, could not be written synonymously in the active voice, because it would require an AGENT Frame Element to be in subject position which is not available. In any case, the syntactic variations among which a generator could choose are of course constrained by the English grammar, thus the fourth example has to be dismissed as invalid.

The second choice a generator has to make is therefore how to map the available Frame Elements to syntactic structures of a sentence. For this choice it has to have information which options are possible for such a mapping given a specific Frame with the available Frame Elements and a lexical unit that evokes the Frame.

The three examples above are very simple sentences, each of which evokes only one or two Frames. More complex sentences, however, typically contain a larger number of Frames. Being able to combine multiple Frames into a single sentence is especially useful in the context of summarization where the key task is to condense information. The ability to verbalize multiple Frames in one sentence is therefore desirable for our NLG system. This implies a third choice for a generator, namely which Frames to combine and in which way.

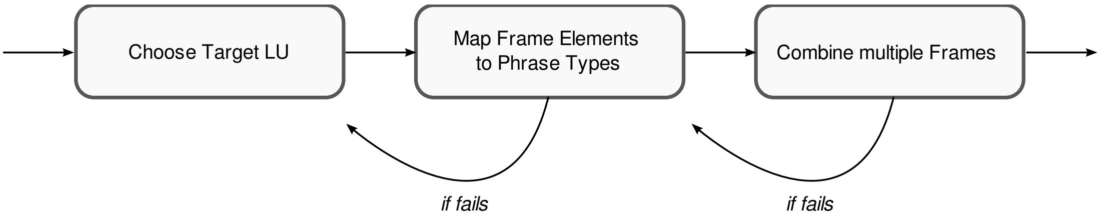


Figure 5.10: Three choice points for the Frame-based generator

The above analysis sets up the primary requirements for a sentence generation component that takes as input the specification of summary contents as Frames. It implies a sequence of three choice points. Whenever a choice made in a previous step conflicts with the options in a subsequent step, the generator may backtrack and choose a different value. The sentence generation is successful if the generator progresses through all three steps, and fails if no more backtracking option is available.

In the following section we introduce a corpus-based approach to address these requirements.

Target LU Determination

The target for a Frame instance that is to be generated does not necessarily have to be determined in all cases. Often times, it is already present in the input structures and can be readily used by the generator. However, when one of the downstream components in the above pipeline fails, backtracking to the first box in order to change the provided lexical unit could be a possibility to find a generator solution for the input, which otherwise would have to be rejected.

The following algorithm makes use of the WORDNET database to generate alternative lexical units, provided an initial one. It works by iteratively trying out all WORDNET synsets the given LU participates in. A synset contains other words with the same meaning, but because the given LU may potentially have multiple meaning, the algorithm sees whether some of the contained words are already known to evoke the same Frame. This information is available in FRAMENET. The synset containing the greatest percentage of words that evoke the same Frame is returned. The words from that synset can be used as alternatives to the given LU.

```
function find_alternative_lus(frame, lu, wordnet)
  synsets := get_synsets(lu, wordnet)

  best_synset := null
  best_ratio := 0

  for each synset in synsets
    count := 0

    for each word in synset
      if evokes(word, frame)
        count := count + 1

    ratio := count / size(synset)
    if (ratio > best_ratio)
      best_ratio := ratio
      best_synset := synset
```

Frame

SUCCESS_OR_FAILURE

Definition

An Agent has attempted to achieve a Goal, and the actual outcome of the Agent's action has been resolved, so that it either specifically matches the Agent's intent (e.g. success) or does not match it (e.g. failure).

Frame Elements

Agent *(Sentient)*:	The Agent makes an attempt to achieve a Goal.
Goal *(Goal)*:	The Goal is what the Agent attempts to achieve.
Role *(Goal)*:	A participant function in a particular event or in events of a particular kind.

Lexical Units

failing.n, failure.n, fail.v, manage.v, miss.v, pull off.v, succeed.v, successful.a, success.n, unsuccessful.a

Figure 5.11: The SUCCESS_OR_FAILURE Frame as defined in the FRAMENET corpus.

```
    return best_synset
```

A similar method is proposed by De Bleecker [2005], but the advantage of our algorithm is that it is able to determine lexical units that have not yet been added to FRAMENET.

Partial Syntax Trees

FRAMENET defines a concrete set of Frames based on the Frame Semantics formalism. In addition to that, it also collects for each Frame a list of lexical units that evoke that Frame. For instance, the Frame SUCCESS_OR_FAILURE describes the situation where an agent has attempted to achieve a goal, and the actual outcome of the agent's action has been resolved (see Figure 5.11). It defines the Frame Elements AGENT, GOAL, and ROLE, and lists the following lexical units to be evoking that Frame: *failing.n, failure.n, fail.v, manage.v, miss.v, pull off.v, succeed.v, successful.a, success.n, unsuccessful.a*[5]. A third source of information FRAMENET provides and which we have not made used of so far is a rich set of sample annotations. In

[5] Note that FRAMENET does not claim this list to be comprehensive for English.

the following we will show how the information from the FRAMENET corpus can be leveraged for text generation.

As we have discussed above, knowing the Frame and the Frame Elements which participate in a specific situation is not sufficient to generate a sentence because the choice which lexical unit is used as the frame evoking one cannot be automated in the general case. We can easily see this with the SUCCESS_OF_FAILURE Frame. Both *fail.v* and *succeed.v* evoke SUCCESS_OR_FAILURE, yet the two verbs describe rather opposing circumstances. Consequently, a choice of the candidate lexical units has to specified a priori to the generator.

Given a preselected lexical unit we can use it to express a situation which describes a success or a failure, but how do we verbalize the Frame Elements so that we end up with a correct and meaningful English sentence? It is easy to see that different circumstances will require different grammatical constructions, e.g.:

- Active vs. passive verb forms call for different syntactic realizations
- Different lexical units have different valences
- The POS of the lexical unit, e.g., noun vs. verb has implications on its use
- etc.

Ideally, a flexible generator should have a way to generate different variations of the same content. We aim at providing a library of generation constructs that the generator can access and from which it can choose alternatives. Building such a library by hand is tedious, costly and error-prone; a method to create it automatically or semi-automatically is thus preferred. But first, we define what exactly we mean by "generation constructs".

For the purpose of illustration, let us revisit the SUCCESS_OR_FAILURE Frame. The sentence *"John managed a feeble smile"* features "managed" as the TARGET, "John" as the AGENT, and "a feeble smile" as the GOAL. Syntactically, we can analyze that sentence as shown in Figure 5.12. The leaves of a syntax tree are the tokens (words and punctuation etc.) of a sentence, their parent nodes denote their parts-of-speech (POS), and all other nodes describe the phrase type of their children. We notice that for each Frame Element there is at least one inner node in the syntax tree that spans the FE, and the same is true for the Target. We call a node which spans the Target of a Frame *Target node* and a node which spans a Frame Element *Frame Element node.* In case there are multiple nodes spanning the same Frame Element or the Target, only the one with the smallest depth is called Frame Element node or Target node respectively.

When we look at the sentence *"John managed a feeble smile."* we realize that the actual content of the Frame Elements are not crucial for the fact that this sentence belongs to the SUCCESS_OR_FAILURE Frame. In fact, if we replaced, e.g., the AGENT "John" with "Mary" or "The designer", we would still yield a meaningful sentence in

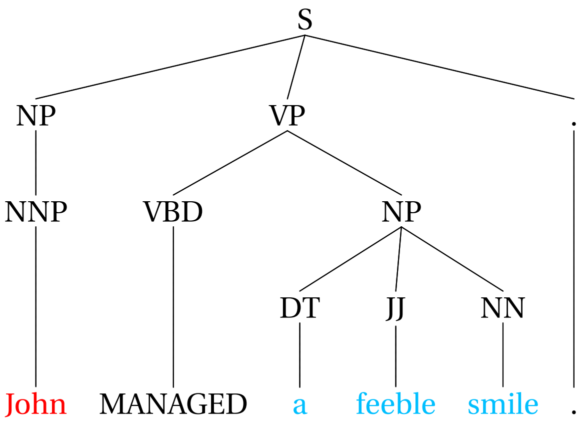


Figure 5.12: A syntax tree for the sentence *"John managed a feeble smile."*

which SUCCESS_OR_FAILURE was evoked. In principle, we could use an existing sentence as a blueprint for novel sentences. But simply replacing words or sequences of words in a given sentence will not always yield plausible or acceptable results, as the following list illustrates:

1. Mary managed a feeble smile.
2. The designer managed a feeble smile.
3. *By the designer managed a feeble smile
4. *Playing soccer managed a feeble smile.

The third sentence replaces "John" with "By the designer". The result is syntactically incorrect, since we expect a noun phrase (NP) in subject position for this particular sentence, not a propositional phrase (PP). From this error we can see that we have to pay respect to certain syntactic constraints when generating a sentence.

The fourth sentence puts the NP "Playing soccer" in subject position, thus is syntactically correct, but nevertheless nonsensical. The SUCCESS_OR_FAILURE Frame lists the type of the Frame Element AGENT as *Sentient*, and "playing soccer" is an activity, not a sentient. We conclude that a generator also has to respect certain semantic constraints. The fillers for the Frame Elements are outside the control of the generator, as they are part of its input data. If a certain Frame is to verbalized we thus expect that the provided Frame Elements align with their definitions and semantic types.

The fact that variations of our original sentence require the AGENT FE to be realized as a noun phrase in order to be syntactically correct becomes evident when we take another look at Figure 5.12, where we see that "John" is spanned by an NP. If

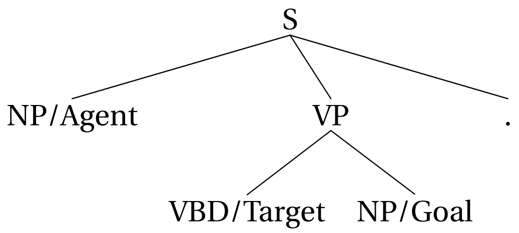
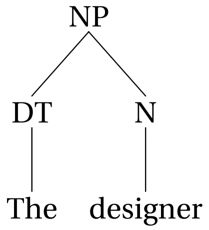


Figure 5.13: A partial syntax tree for the Frame SUCCESS_OR_FAILURE and the Target lexical unit *manage.v.*

we keep the syntax tree identical (except for the leaves), we can be certain that variations of the original sentence will be syntactically correct. This is what we observe in sentences 1 in the above list which is analyzed syntactically in analogy to Figure 5.12. However, at the same time we note that some changes are possible that keep the basic spirit of the syntactic analysis intact but allow for minor variations in the syntax tree. For instance, in the above sentence 2, we use a definite noun phrase for AGENT instead of a proper name. Thus the derivation looks like this:



Likewise, if the sentence ended in an exclamation point or a question mark instead of a period, we would still judge it correct and it would still evoke the SUCCESS_OR_FAILURE Frame. All in all, we can claim a syntactic realization of that Frame to be valid, if the upper part of the syntax tree looks like the one in Figure 5.13. Here, we have cut the syntax tree of *"John managed a feeble smile."* to a certain subtree and annotated the Target node and the Frame Element nodes with their respective roles. We call the resulting tree a *partial syntax tree.*

A partial syntax tree represents the essential valences of a frame evoking lexical unit. It encodes one way of realizing a Frame, given a Target LU and a set of Frame Elements. A generator can use these tree elements – a partial syntax tree, a frame evoking lexical unit, and a set of Frame Elements to generate a full sentence if the following conditions hold:

1. The partial syntax tree has *S* as its root node.

2. There is a mechanism that can generate each Frame Element as the phrase type or POS specified by the partial syntax tree.

Creating an English sentence with this setup can be realized by starting with a partial syntax tree which the generator then expands by iterating over all leaf nodes. If the leaf node is of the form X/Y with X being a part of speech or a phrase type and Y being "Target" or one of the Frame's Frame Elements, the above mechanism for generation the correct phrase type is invoked. At the end, all leaves from the original partial syntax tree have been extended, and the partial syntax tree is now a full syntax tree. The sequence of leaf nodes is the surface form of the generated sentence.

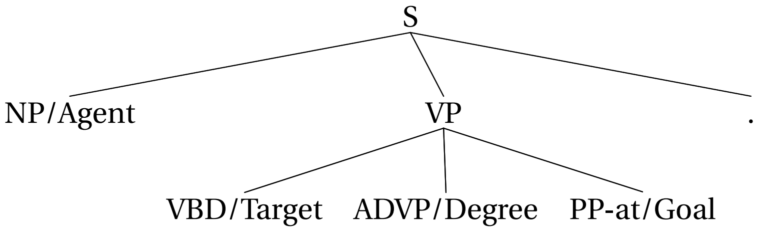
For instance, consider the following input specification:

Input Specification

Frame	Success_or_failure
Target LU	fail.v
Frame Elements	• A political group (Agent) • Getting a certain bill passed (Goal) • Miserable (Degree)

and the following excerpt from an assumed library of partial syntax trees which the generator has access to:

Generator Library

Partial Syntax Tree	 S NP/Agent VP . VBD/Target ADVP/Degree PP-at/Goal

The above input structure contains three Frame Elements, Agent, Goal, and Degree, the latter being a *non-core* Frame Element, since the degree of the failure is not essential to the concept of failure.

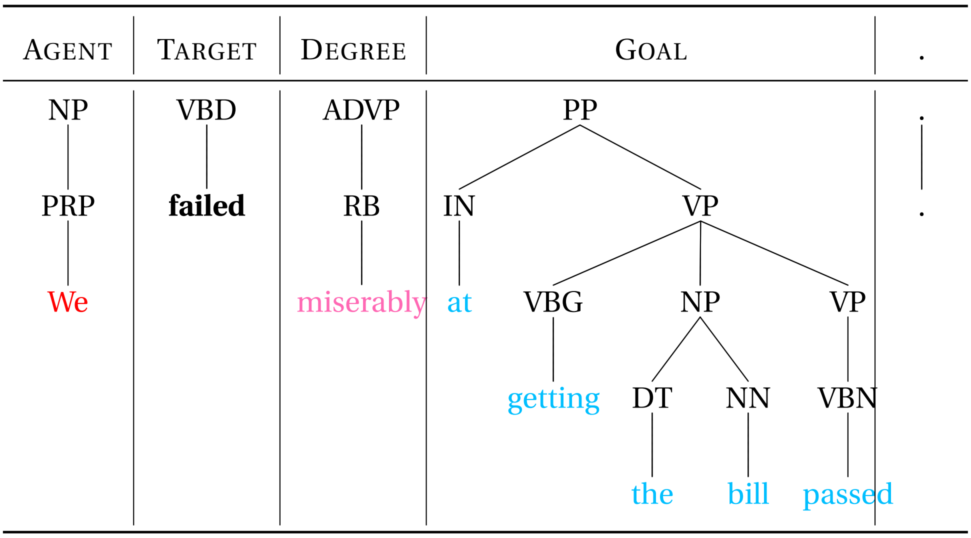
The generator can make use of the partial syntax tree to generate a complete sentence for the given input specification. The partial syntax tree contains five leaf nodes which have to be extended:

- NP/Agent
- VBD/Target
- ADVP/Degree

- PP-at/Goal
- .

If one of these leaves cannot be extended, the generation of the sentence fails. For instance, if the Target LU were the noun *failure.n*, it would not be possible to extends the VBD/Target node since VBD is the POS tag for "verb, past tense" (see Appendix B). In that case, the generator would have to find another partial syntax tree in its library and attempt generating a sentence from there.

In this concrete example, however, it is possible to extend the Target and the Frame Elements:



The concrete realization of a Frame Elements depends on the domain and the discourse context. The Frame Element AGENT is verbalized here by the pronoun *we* which we assume could be understood in a specific context as referring to the political group in question. We make a similar argument for the use of the definite article "the" with *bill*, assuming that the reader of the generated sentence would understand from context which bill is referred to. Such domain-dependent decisions are out of the scope of the core generator. We thus argue for a generation architecture that relies on a domain-specific component to generate the Frame Elements.

Let us define a generator that works as described more formally.

Definition A *Frame Generator* is a tuple $(L, T, f, ...)$ where L is an alphabet of lexical units, T is a set of partial syntax trees, f is a function that maps lexical units to partial syntax trees.

Note that the tree does not only contain leaves that belong to the Frame structure, i.e. Target node and FE nodes, but that the sentence ending punctuation is kept as well. However, the concrete leaf – in this case a period – is in fact cut out, and only the POS maintained which can be extended to either a period, an exclamation point, or a question mark. Questions in English often feature a different syntax than statements (word order, use of modal verbs), and should thus be handled with care, both when creating a partial syntax tree from a concrete example question and when using a partial syntax tree in generating a question.

Which nodes from a full syntax tree are kept and which ones are dismissed in the general case is defined by a construction algorithm below.

Leveraging the FRAMENET Corpus

The example above assume the existence of a library of partial syntax trees for generation. Such a library can be generated automatically from annotations such as the ones provided in the FRAMENET corpus.

Let F be a Frame and s a sentence in which the F is evoked. If S is a syntax tree of s then we derive the partial syntax tree $T = (V_T, E_T)$ as a subtree of S by defining V_T as follows. A node n is in V_T if and only if:

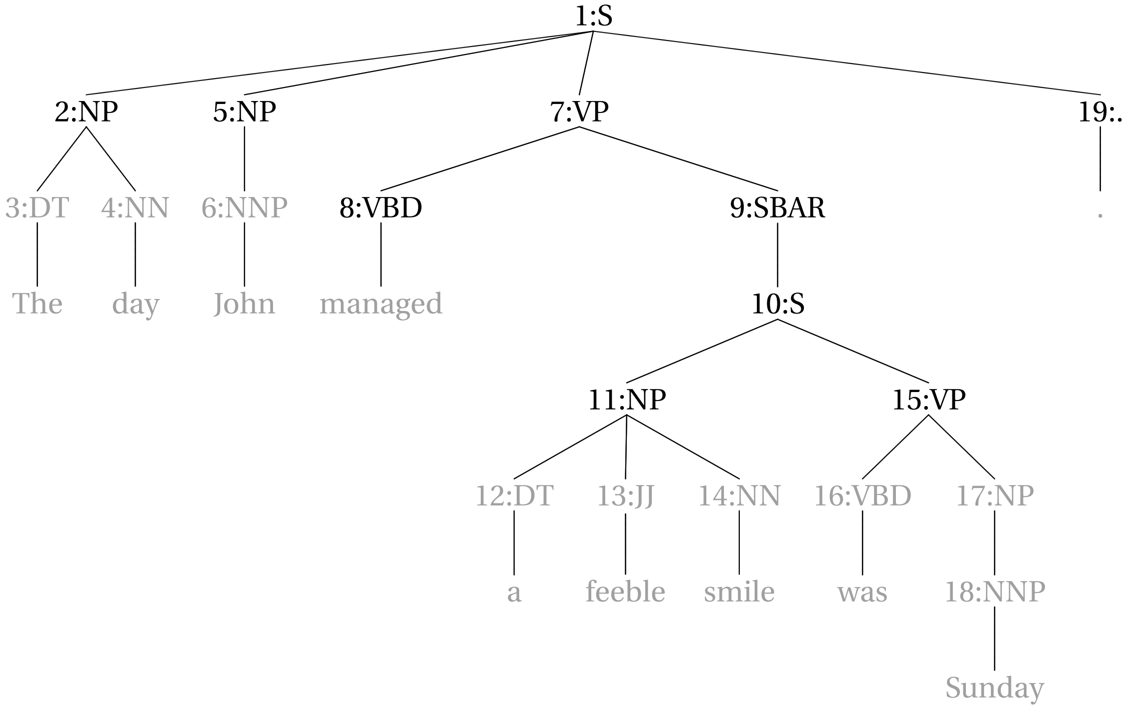
1. n is the Target node or a Frame Element node or
2. n's sibling is in V_T
3. n has a descendant d_1 that is in V_T and
    a) n has an ancestor that is in V_T or
    b) n has a descendant d_2 that is in V_T and $d_1 \neq d_2$

Let's illustrate this algorithm with a variation of the above example sentence:

The day John managed a feeble smile was Sunday.

Here, the SUCCESS_OR_FAILURE Frame is embedded into a more complex syntactic structure, which we can analyze as follows[6]:

[6] We enumerate the inner nodes solely for the purpose of referencing them.



Such an analysis can be produced automatically using a syntactic parse, such as, e.g. [Klein and Manning, 2003]. The recursive definition of the partial syntax tree first includes the target and Frame Element nodes per rule 1. These are: 5, 8, 11. Because of rule 3b, nodes 1 and 7 also belong to the partial syntax tree. Rule 3a includes nodes 9 and 10. Finally, because of rule 2, nodes 2, 15, and 19 complete this particular partial syntax tree for the SUCCESS_OR_FAILURE Frame, which discards the nodes with gray labels.

Constructing partial syntax trees in this fashion means that some of the leaf nodes of the result will be Frame Elements while others sometimes will not. For instance, in our example nodes 5 and 11 are Frame Element nodes, node 8 is a target node, and the remaining leaf nodes neither Frame Element nor target nodes. We call the leaves that are not Frame Element or target nodes *free nodes* and define: a partial syntax tree is *saturated* if it does not contain any free nodes.

In order to generate a sentence from a partial syntax tree, it has to be saturated. The FRAMENET corpus contains 484 distinct Frames, but for only 379 of them (78.3%) we find at least one partial syntax tree in the corpus that is saturated. Of these, 111 have an S node (sentence node) as the root type. That means that for 373 Frames, we don't have a partial syntax tree to generate a sentence from. The reason for this is that the annotations of the FRAMENET corpus are based on material from the British National Corpus (BNC) which consists mainly of sophisticated publications with complex sentence structures. Consequently, the annotated sentences typically contain information from multiple Frames at once. If we extract a partial syntax tree for one of the Frames, it is likely to span only a certain part of the sentence and the partial syntax tree extraction algorithm will cut it down to the smallest spanning phrase. Thus the root of the partial syntax tree will have a phrase type different from S.

One way to fix this is to manually add saturated partial syntax trees for the 373 Frames for which we can't derive such partial syntax trees automatically. A second remedy would be to add simple phrase structure rules of the kind *S* → *NP VP* which the generator could use to derive complete syntax trees. In practice however, this problem arises less often because most sentences are made up of more than one Frame.

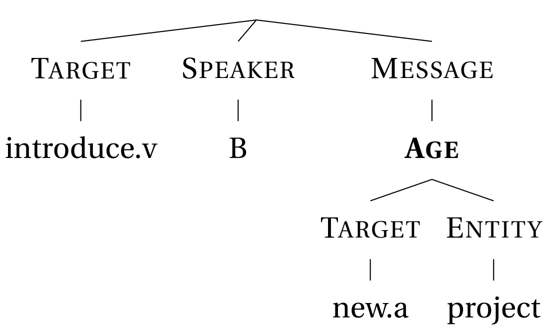
Multi-Frame Sentences

The previous section introduced partial syntax trees, and how they can be used as templates to generate English text. The result of the generation is a phrase, which may be a full sentence (if the root node of the used partial syntax tree is *S*) or any other phrase type. The generation process consist of enriching the free nodes of a partial syntax tree with the information found in the framal input specification for the sentence.

However, as we know from chapter 5.4, a sentence usually encoded more than just one Frame. A realistic generator thus must provide means to integrate multiple Frames unless the application scenario only calls for very simple, one-Frame sentences.

The way we represented the propositional content of sentences in previous chapters is that of a Frame dependency tree. In this section, we extend the partial syntax tree approach to handle such trees and show how to generate complex English sentences from them.

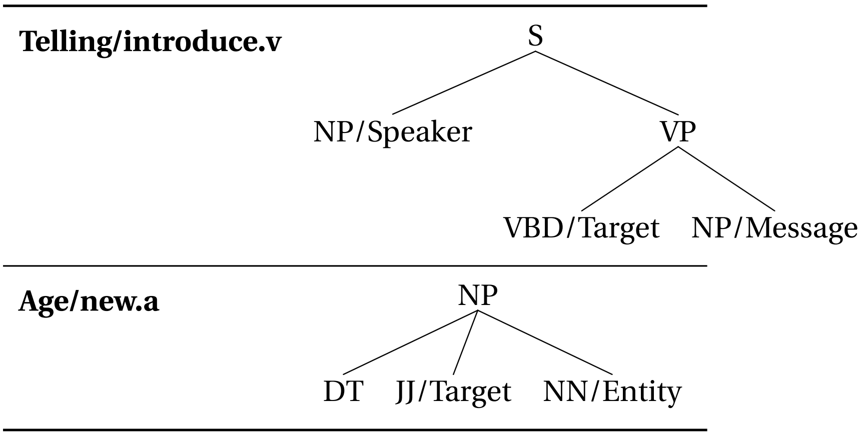
For the sake of illustration, consider an input specification consisting of two dependent Frames:



The input also contains the following entity mapping:

B	→	entity1
the project manager	→	entity1
the project	→	entity2

In the generator's library, we find the following two partial syntax trees:



With this, the generator can create an English sentence with the following procedure:

1. Find a partial syntax tree for the root Frame with *S* at the root.
2. For every Frame Element containing an entity reference, fill them into the partial syntax tree as before.
3. For every child Frame of the current Frame, find a partial syntax tree. Let *pt* be the phrase type at the root of that tree.
4. Find all occurrences of child's *pt* in the current partial syntax tree.
5. Try to find a downward mapping of all of child's nodes to the current occurrence of *pt* in the parent tree.
6. Make child the current Frame and recursively generate it starting from step 2.

We call this process *patching* because it overlays two partial syntax trees so that a new partial syntax tree emerges. The crucial point in this algorithm is step 5. We now define what we mean by *downward mapping* by giving a construction procedure for it. The basic idea is that two partial syntax trees can be patched together when their inner node structure lines up.

1. Let n_1 and n_2 be nodes in two partial syntax trees. A downward mapping f is constructed as follows.

2. If n_1 and n_2 are the same phrase type add $n_1 \rightarrow n_2$ to the mapping, otherwise the procedure aborts.

3. If n_1 or n_2 is a leaf node, the current mapping is returned.

4. Otherwise, recursively extends the mapping by processing the i-th child of n_1 with the i-th child of n_2.

5. If the number of child nodes differs for n_1, n_2, the procedure aborts.

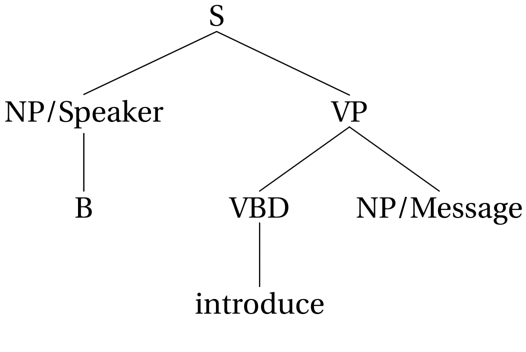
In addition to the purely syntactic mapping, we also require two semantic constraints:

- If the downward mapping maps two Frame Element nodes together, they must refer to the same entity.

- Let *fe* be the Frame Element of the parent Frame under which the dependent child Frame is listed. Then the downward mapping is only valid if the child's target node is mapped as a descendant of *fe*.

Let us illustrate this procedure with the example structures introduced before. The procedure begins with the TELLING Frame because it is at the root of the dependency tree. The library of partial syntax trees is consulted and the one given above is retrieved. Note that it has an *S* node at the root.

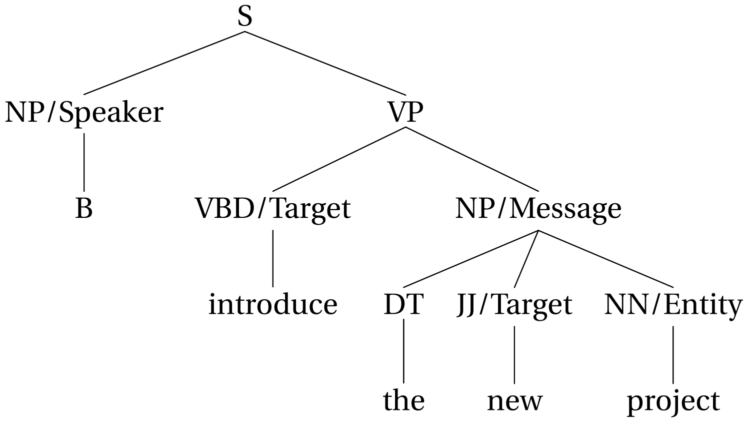
The target and the Frame Element SPEAKER can be filled in as in the case of single Frame sentences in the previous section.



The Frame Element MESSAGE does not reference an entity, but a dependent Frame, namely AGE. The partial syntax tree library contains a partial syntax tree with NP at the root (see above). There are two occurrences of NP nodes in our intermediate representation, thus we have two candidates for a downward mapping: NP/Speaker and NP/Message.

For NP/Speaker, the downward mapping fails because the child nodes of that node and the root of the AGE's partial syntax tree do not match: the former's child is THE PROJECT MANAGER while the latter's first child is DT.

The candidate NP/Message, however, matches and it also complies with both of the semantic constraints, because AGE's target node is mapped as a child of NP/Message. Thus we can combine the two partial syntax trees in the one using the downward mapping and yield or result tree:



Surface realization

For the SPEAKER FE, the input structure contains the reference *B*. By consulting the entity mapping we see that reference *B* identifies entity1, which also has a second reference available, *the project manager*.

Strategies for choosing among multiple references should consider, among other things:

Descriptiveness When an entity is first mentioned, a more descriptive realization might be desirable. In subsequent mentions, however, less verbose realizations could be preferred.

Repetition If an entity appears multiple times in the text, using the same reference again and again can become repetitive and unpleasant to read. One way to address this problem even for entities that only come with a single reference, is the introduction of pronouns.

Phrase type matching The partial syntax tree requires a Frame Element to be realized as a specific phrase type. Only those references that match can be used for realization (e.g. compare *walk* versus *walking*).

Writing style The realizer should only choose references that are appropriate and match the writing style of the rest of the sentence.

Since the references in our system emerge as the result of the interpretation of the transcript and the transformation of this interpretation, they are not always in lemma form. In contrast, the partial syntax library always stores the targets as lemmas. It can therefore become necessary to adjust the morphology of the different references. This is a complex topic in itself. In the present approach, we merely address subject-verb agreement by inflecting verbs according to the number of the governing subject.

In our example, We opt for *project manager* because it is more descriptive, but depending on the context, simply using *B* might be an equally good or better option. After addressing the inflexion of *introduce*, the final generated sentence is:

The project manager introduces the new project.

5.7 Example of a System Run

We now give a detailed example of the processing steps of MEESU, as outlined above. In particular, we show the inputs to each of the three main processing stages, and how the summarizer transforms them into the output which are then passed to the next stage.

Interpretation of the Transcript

For that we consider an excerpt of meeting ES2008a from the AMI corpus, which belongs to the so-called *scenario* meetings of the corpus (see Section 4.4). In this meeting, a new team meets for the first time to work together on the new project of creating a remote control together. The four participants are the project manager (A), an industrial designer (B), a user interface designer (C), and a marketing expert (D).

This information about the participants as well as the fact that they are a team and that they are participating in a meeting is reflected in the initial state of MEESU's *entity mapping* before processing begins (Table 5.3). Here, we have an entity for every participant, one for the group, and one for the meeting. Although the speakers do not refer to each other by the ID letters, these references are inserted in the automatic mapping of dialog acts to Frame instances, as we will see below. The entity representing the participants as a group, contains three different synonyms. This is to allow some variation in the text generation component later on. Also, we assume by default that when the meeting participants use the word "we", they refer to the meeting group.

The following is a transcript of the beginning of the meeting. It is a manual transcript in which some speech disfluencies have automatically been corrected with the GRODI tool.

Table 5.3: Initial state of MEESU's entity mapping. The internal speaker ID's are mapped to different entities–shown as boxes–which contain verbal descriptions of the participant's roles. There are also entities representing the members as a group and the meeting itself.

Reference		Referent
A	→	*the project manager*
B	→	*the industrial designer*
C	→	*the user-interface designer*
D	→	*the marketing expert*
we	→	*the team, the group, the meeting participants*
ES2008a	→	*the meeting*

1 **D:** Hmm.
2 **A:** Okay.
3 **A:** Good morning everybody.
4 **A:** Um I'm glad you could all come.
5 **A:** I'm really excited to start this team .
6 **A:** Um I'm just gonna have a little PowerPoint presentation for us, for our kick-off meeting.
7 **A:** My name is Rose Lindgren.
8 **A:** I'll be the Project Manager.
9 **A:** Um our agenda today is we are gonna do a little opening
10 **A:** and then I'm gonna talk a little bit about the project,
11 **A:** then we'll move into acquaintance such as getting to know each other a little bit, including a tool training exercise.
12 **A:** And then we'll move into the project plan,
13 **A:** do a little discussion
14 **A:** and close,
15 **A:** since we only have twenty five minutes.

16 **A:** First of all our project aim.
17 **A:** Um we are creating a new remote control which we have three goals about,
18 **A:** it needs to be original , trendy and user-friendly.
19 **A:** I'm hoping that we can all work together to achieve all three of those.

This is the input to Meesu's interpretation component. The first steps of interpretation is labeling each utterance with a dialog act and analyzing the Framal content of each utterance. For the former task, we use the gold-standard annotation available in the AMI corpus while the latter task is performed with the SEMAFOR parser. The result is shown in Figure 5.14. The dialog act labels are printed in green in the middle column, the recognized Frames in the rightmost column.

Besides the name of the each Frame, SEMAFOR also outputs which word in an utterance evokes the Frame, i.e. the Frame's target, and any recognized Frame Elements. For instance, the full annotation of utterance 4 is given in Figure 5.15, that for utterance 7 in Figure 5.16. Note that while SEMAFOR's analysis of utterance 4 is quite accurate, it sometimes misinterprets contents, as e.g. in the case of utterance 7. Here, the name *Rose* is mistaken as a past tense form of the verb *rise*, which evokes the Change_posture Frame. This is an example for an incorrect analysis of a target LU. Also, the Frame Element analysis of the other Frame, Being_named, is not quite correct. The Frame Element Name should have been analyzed as *Rose Lindgren.* Here, however, the target LU *name* was recognized correctly as evoking Being_named. A third type of mistake in the analysis is failing to recognize certain Frame Elements. For *Being_named*, the word *my* for instance, should have been analyzed as the Frame Element Entity, to encode that the speaker talks about her own name.

Next, the dialog acts are mapped into a Framal representation as well, as specified in table 5.2. Each of the resulting Frames has two Frame Elements, encoding the speaker of the utterance and the full content. According to that table, if the dialog act label is one of *bck*, *be.pos*, or *be.neg*, no Frame mapping is defined and the utterance is thus not further interpreted. This is the case e.g. for utterance 4.

The target and Frame Element spans of each of the Frames in these analyses define the dependency relation described in Section 5.4 which induces a graph representation. For every utterance in the transcript, the Frame derived from the dialog act label becomes the root node of a tree of Frames. A Frame becomes a descendant of another Frame when the target span of the former lies inside one of the Frame Element spans of the latter. The final representation of utterance 7 is thus as shown in Figure 5.17. The output of the interpretation phase is a Frame tree for every utterance of the transcript.

1	oth	-
2	stl	Desirability
3	be.pos	Calendric_unit, Desirability
4	be.pos	Emotion_by_stimulus, Capability, Arriving
5	be.pos	Emotion_directed, Aggregate, Activity_start
6	off	Possession, Quantity, Discussion
7	inf	Being_named, Change_posture
8	inf	People_by_vocation, Project
9	inf	Openness, Temporal_collocation, Intentionally_act
10	inf	Quantity, Statement, Temporal_collocation, Project
11	inf	Awareness, Required_event, Gizmo, Inclusion, Education_teaching, Motion , Practice, Increment, Temporal_collocation, Quantity
12	inf	Project, Temporal_collocation, Project, Motion
13	inf	Quantity, Intentionally_act, Discussion
14	inf	Social_connection
15	inf	Cardinal_numbers, Cardinal_numbers, Possession, Calendric_unit, Sole_instance
16	inf	Purpose, Quantity, Project, Ordinal_numbers
17	inf	Relational_quantity, Creating Be_in_control, Purpose Cardinal_numbers, Familiarity, Possession
18	inf	Required_event, Trendiness
19	be.pos	Accomplishment Possibility Being_employed Desiring Quantity Collaboration

Figure 5.14: List of Frames as output by the SEMAFOR parser for each utterance in the transcription excerpt.

Transformation into summary contents

The next step in the processing is the application of macro-rules to transform the list of Frame trees which are output by the interpretation phase into a representation of summary contents. The transformation phase proceeds by iterating over that list and applying construction, abstraction, and deletion rules in turn.

Frame	Emotion_by_stimulus
Target LU	glad
Frame Elements	• I (Experiencer) • glad you could all come (Event)

Frame	Capability
Target LU	could
Frame Elements	• you (Entity) • come (Event)

Frame	Arriving
Target LU	come
Frame Elements	• you (Theme)

Figure 5.15: Output of SEMAFOR for utterance 4: “Um I’m glad you could all come.”

Construction

Construction rules make use of a scripts knowledge base. The first time one of these scripts is triggered in the course of our example is in utterance 7. Meesu’s script base contains, among others, the script shown in Figure 5.18.

In the context of a kick-off meeting, informing the meeting participants about one’s name is transformed into an act of introducing oneself. If the Frame structure listed as *scene* in the script is detected in the interpretation, it triggers the generation of a new Telling Frame that encodes that the speaker introduced him- or herself to the meeting group. Ideally, we would require that Being_named has a Frame Element Entity that must refer to the same entity as Speaker in Telling. However, as we have seen, we cannot rely on SEMAFOR to always produce a perfect analysis. Thus we suggest a more lenient variant of the script in which such a constraint is not imposed.

In the transcript excerpt considered here, utterance 7 matches the scene of the IntroduceSelf script. Because this script contains only a single scene, a match thereof is sufficient for the construction of the script’s Telling Frame with Target *introduce.v*. The result is shown in (Figure 5.19).

Frame	Being_named
Target LU	name
Frame Elements	My name (Name)
Frame	Change_posture
Target LU	Rose
Frame Elements	-

Figure 5.16: Output of SEMAFOR for utterance 7: “My name is Rose Lindgren.”

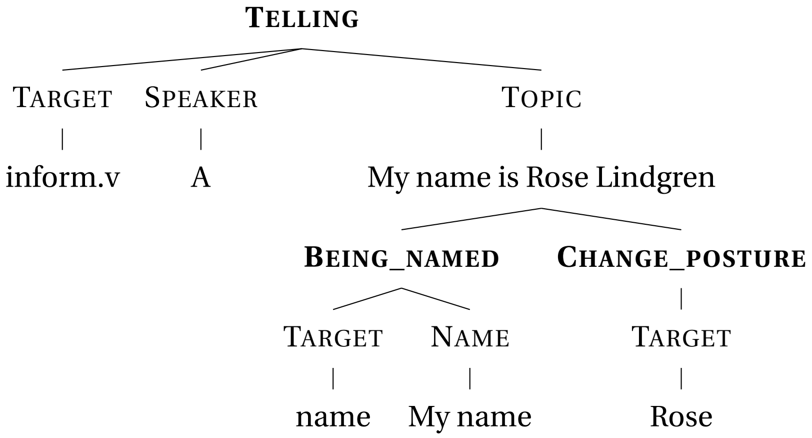


Figure 5.17: Final interpretation of utterance 7: “My name is Rose Lindgren.”

SEMAFOR also detects three other instances of Being_named in the transcript, in lines 51, 56, and 88:

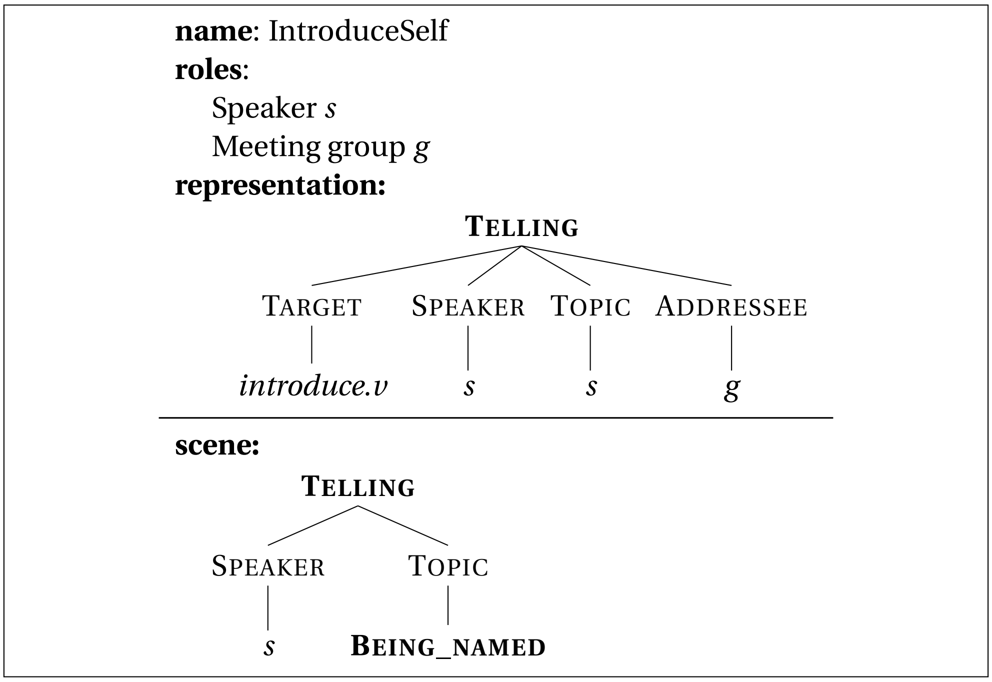
51 **B:** My name is Alima Bucciantini.

56 **A:** How do you spell your name?

88 **C:** my name is Iain

For lines 51 and 88, the IntroduceSelf script produces instances of Telling the same way as for line 7 before. Line 56, however, does not trigger the script. The reason for that is that the interpretation phase outputs the Frame shown in Figure 5.20.[7] The root node of this interpretation is Request rather than Telling, there-

[7] SEMAFOR also wrongly labels the word “do” in line 56 as evoking Intentionally_act, which



name: IntroduceSelf
roles:
Speaker *s*
Meeting group *g*
representation:

scene:

Figure 5.18: The *IntroduceSelf* script. It encodes that introducing oneself consists of saying one's name.

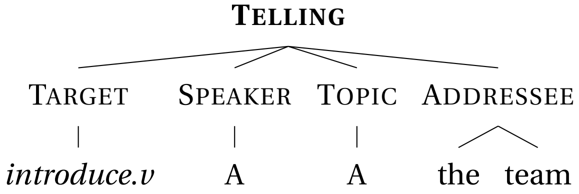


Figure 5.19: The result of applying the *IntroduceSelf* script to the Frame tree of Figure 5.17

fore IntroduceSelf does not match.

The construction component of MEESU is passed one Frame tree at a time. It then checks all scripts in the knowledge base for structural matches of the encoded scene trees with the input tree. In general, a script can require more than one scene, thus partial matches of sub-scenes are stored and subsequent Frames may substantiate previous partial matches until a complete match has been reached. To do so, the construction component checks every Frame input tree against every contained

has been omitted from this tree for brevity.

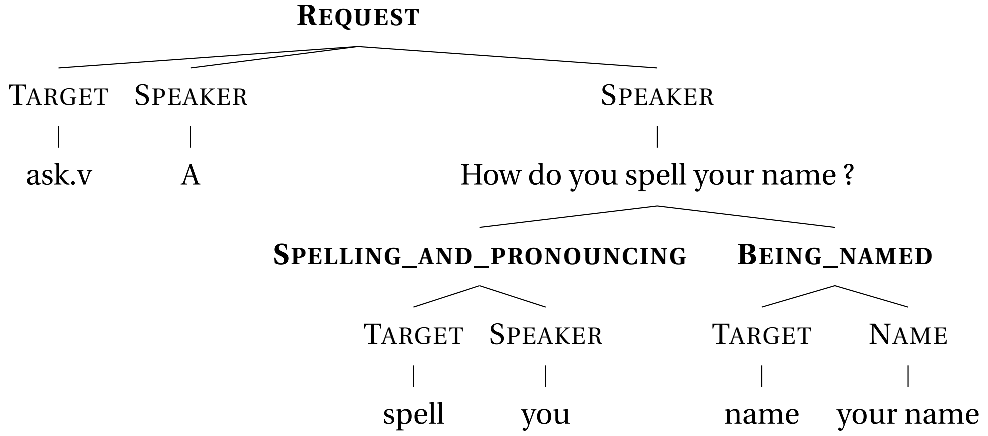


Figure 5.20: Part of the final interpretation of utterance 56: "How do you spell your name?"

scene description in all of the scripts, and stores all successful partial matches. Only when a match has been found for each of the contained scenes will the constructor create a new instance of the script Frame. An example for this is shown in Figure 5.21. The AnnouncedMonolog script describes the effect that a speaker in a meeting will sometimes introduce a longer contribution by giving a short description beforehand. It consists of seven sub-scenes. First, the COMMUNICATION Frame is found in a transcript when a *stall* dialog act occurs. This is usually a contribution without any real content, but we find that it often marks a topic change. The second or the third scene is the actual announcement of a topic, that is, the speaker says what he or she is going to talk about. The reason for allowing an optional second scene is that changes of topic are sometimes begun with a meta remark about changing the topic. The announcement then has to be substantiated in further scenes. We require at least four subsequent TELLING instances after scene 2. This is to distinguish announced monologs from e.g. stating the agenda at the beginning of a meeting. In other words, it is not sufficient that a speaker announces a monolog, the monolog then actually has to happen right away after the announcement.

For a script to be triggered, further constraints may apply besides the matching of the Framal structure of scenes. For instance, a script may require that certain Frame Element refers to the same (or *not* to the same) entity as some other Frame Element. As we have mentioned before, enforcing such constraints too strictly may harm the performance when the output of the interpretation phase is slightly imperfect. Other constraints are requiring a certain temporal gap between script scenes, or requiring the input Frames that match the scenes follow directly one after another in the meeting interpretation, without interruption by other Frames. This is

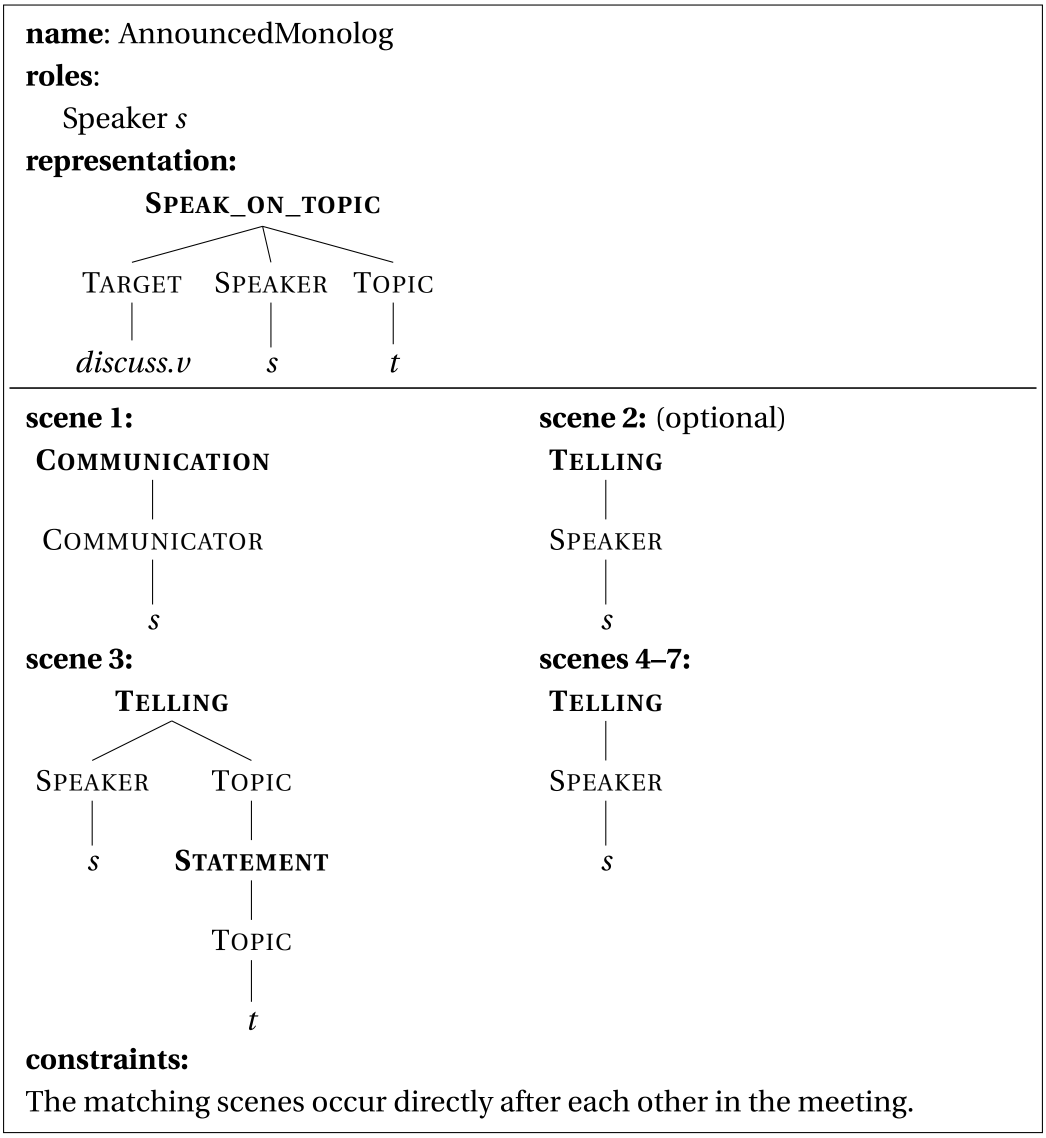


Figure 5.21: The *AnnouncedMonolog* script.

the case in the AnnouncedMonolog script, in order to assert that what follows the announcement scene 2 is indeed a monolog.

The AnnouncedMonolog script is not matched in the excerpt given above, however, it appears a little later in the same meeting:

168 **A:** Okay,
169 **A:** moving on to slightly more serious stuff.

170 **A:** We're gonna talk about project finances.

171 **A:** Um we have a couple

172 **A:** we'd to sell it for about twenty five Euro with the profit aim of fifteen million Euro from our sales

173 **A:** and because this is such this is for television it's a we have a market range of Internet, it's an international market range,

174 **A:** we don't have to worry about specifics.

The Frame instance that results from the AnnounceTopic script is shown in Figure 5.22.

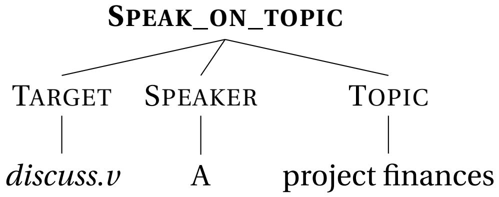


Figure 5.22: The Frame instance created by the application of the AnnounceTopic script on lines 168–174 of the transcript of meeting ES2008a.

All Frame instances created by the application of a script are collected in a list and returned by the construction component. Likewise, the Frames that triggered a script application by matching its scenes are collected in a second list which is later on passed to the deletion operation. The rationale here is that if a certain Frame tree has contributed in the construction of another Frame instance, we consider the latter to already contain the information of the former to a certain degree. Thus we avoid generating redundant information by putting the scene Frames into a delete list.

Abstraction

After the construction component has finished processing a given utterance analysis, the same analysis is passed to the abstraction component. In addition, all Frame trees generated by the construction step are also passed to the abstraction component in sequence.

As described above, the abstraction component derives new Frame instances that are logically entailed by others. One way it achieves this is by making use of the information-retaining transformations. Reconsider for instance the BEING_NAMED Frames of utterances 7, 51, and 88 from the transcript which the construction component transformed into three TELLING Frames via the IntroduceSelf script.

The abstraction component detects the parallel structure of the TELLING Frames. This is the case when the Frame target are the same and when each contained

Frame Element either refers to the same entity across all instances, or refers to the speaker of Frame in question. In such case, the abstraction component first creates a new group entity and then a new instance of the Telling Frame. The new Frame instance contains the same target and the same Frame Elements as the original Telling Frames, however, references to the speaker in one of the original Telling Frame Elements is replaced by a reference to the new group entity. In our example, this leads to the new Frame depicted in Figure 5.23.

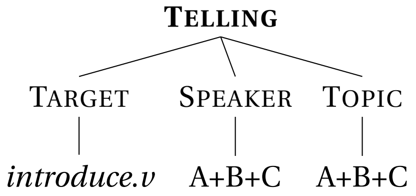


Figure 5.23: The Frame instance created by the abstraction component from three different instances of Telling, licensed by the operation *grouping of entities.*

Our example shows how the output of one transformation component can result in further processing in another one. The Frames created in the abstraction component are returned as a list, and analog to the construction component, the source Frames for the entailment are moved into a delete list. In this particular instance, the construction component produces no further script application and processing continues in line 89 of the transcript.

Deletion

Deletion in Meesu is performed as an operation consisting of two parts. One part of the operation already happens during the interpretation phase, where three types of dialog acts are filtered out from subsequent processing. However, the main part of the operation is implemented in the deletion component.

The deletion component is run after the construction and abstraction components have finished processing the meeting process. If successful, these components have produced new Frame instances in addition to the ones resulting from the interpretation phase. The deletion component acts as a filter on the set of all of these results. In the case of our example, the set of all Frames consists of the Frames listed in Figure 5.14, Figure 5.19 (plus the discussed variations thereof for speakers B and C respectively), and Figure 5.23. Additionally, the construction component creates a Frame from line 16 of the transcript which encodes the project manager's monolog about the project's goal.

Of these Frames, Meesu deletes all but the one produced by the abstractor, and two of the constructor Frames. The Frames that were originally produced by the constructor but then re-used in the abstractor get deleted, too. The deletion component uses two kinds of information for the deletion decisions. First, the delete list which the constructor and abstractor components produce is consulted. If a Frame is contained in that list, even if it was created by one of the other components, it gets deleted. The rationale here is that the constructor and abstractor only put Frames in the delete list, for which they can offer an alternative. This way, the redundant inclusion of similar contents in the abstract is reduced.

A second aspect marks every utterance in the transcript as relevant or not, similar to extractive summarization. In this particular case, MEESU uses the existing relevance annotation in the AMI corpus, but also provides a mode for a Murray's integer linear programming method (see Section 3.6). To make use of the findings from such approaches, the MEESU system only allows Frames that stem from an utterance marked as relevant, or that were produced by using at least one such Frame.

The final output of the transformation phase for the excerpt thus is the one shown in Figure 5.24.

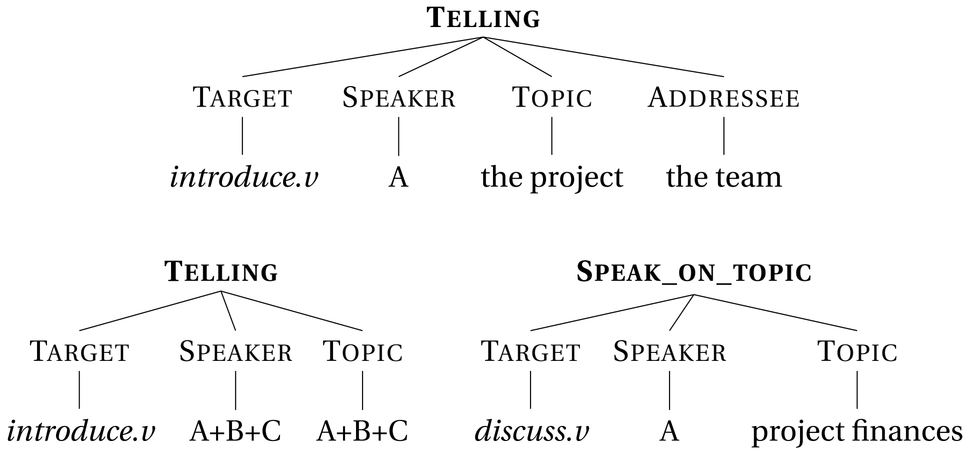


Figure 5.24: The final output of the transformation phase.

Generation of the Summary Text

The final step in MEESU's pipeline is the generation of a summary document. For the parts of the meeting transcript discussed in the previous section, the result of the transformation phase consists of two instance of the TELLING Frame and one of the SPEAK_ON_TOPIC Frame.

These Frames have already a target LU associated with them. However, in the case of TELLING, where both instances have the verb *introduce.v* as their target, a generator could opt to change the LU to reach a more varied summary text. MEESU does not currently implement that feature.

In the next step, the generator component consults the database of partial syntax trees for these two Frames. For TELLING, this database contains 253 different partial syntax trees, for SPEAK_ON_TOPIC 105. However, not all of them match the target LU's and the generator can also only use trees with a sentence node at the root, since we want to generate a syntactically correct, complete sentence. Even with the number of partial syntax trees reduced, the generator may end up with multiple options. It will then select a partial syntax tree that best matches the provided Frame Elements, has the least number of unsaturated leaf nodes (ideally none) and otherwise produces the shortest sentence. In our example, the generator eventually selects the partial syntax trees shown in Figure 5.25.

The Target and Frame Element nodes in these partial syntax trees have to be saturated with actual surface forms. For that, the generator looks up entity references in the entity mapping (cf. Table 5.3). If an entry is found, one of the variations of that entity's lexicalization is chosen, otherwise the reference itself is as a lexicalization. For instance, the Frame Element SPEAKER in the first TELLING instance is associated with the reference "A". The entity mapping maps this to a specific entity which contains the lexicalization "the project manager". Since the phrase type matches (SPEAKER is attached to an NP node in the partial syntax tree), the verbalization succeeds and the Frame Element is realized as "the project manager". Without such an entry in the mapping, the Frame Element would simply be realized as "A". The phrase type match is already considered during the selection of the partial syntax tree. That way, the generator can pick a partial syntax tree that best accommodates the available lexicalization options.

Figure 5.26 exemplifies the final syntax tree of the first TELLING Frame. The syntax trees for both of the other Frames are saturated similarly, but the second TELLING Frame requires additional treatment. Here, the SPEAKER FE and the TOPIC FE both refer to the same compound reference A+B+C. The abstractor has inserted a group entity into the entity mapping that lexicalizes the components through a coordinated conjunction. However, since the same entity appears twice in that sentence, it is nicer to realize one of them as a pronoun. The generator tests for such cases; here, it realizes the Frame Element TOPIC with a pronoun as long as there is no reason to believe that the pronoun could refer to some other entity.

Lastly, the generator applies some basic morphological rules to assert case, number, person and gender agreement between dependent items. For instance, the pronoun chosen to realize the second TELLING's TOPIC Frame Element is "themselves" because the FE refers reflexively to a plural entity. Detection of the relevant morphological features is done with a small number of heuristics rules.

With all Target and Frame Element nodes thus realized, the final step is to output the surface forms at the leaves of the trees. For our excerpt example, the finally produced summary text is:

> *The project manager introduces the project to the team. The project*

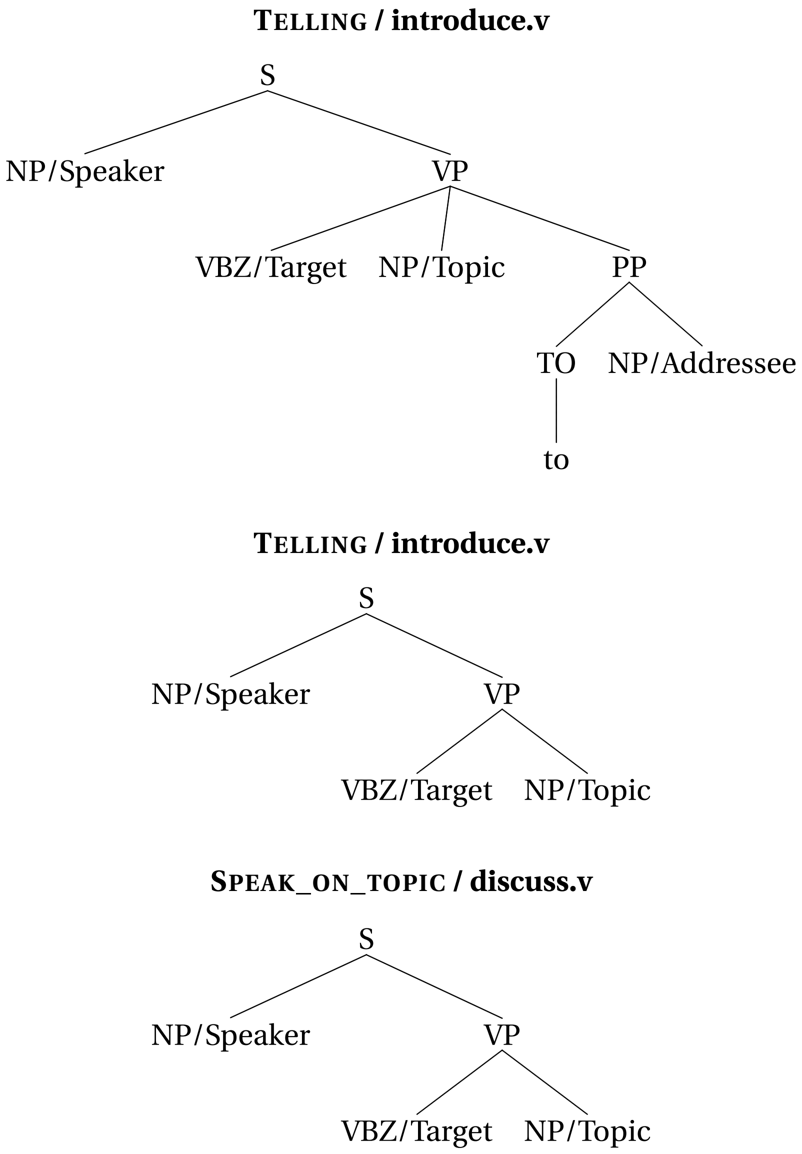


Figure 5.25: The selected partial syntax trees for the three Frame instances.

> *manager, the industrial designer and the user-interface designer introduce themselves to the team. The project manager discusses project finances.*

5.8 Chapter Summary and Discussion

This chapter describes the overall architecture of an abstractive meeting summarizer, and describes in detail the steps involved to represent a meeting, interpret a

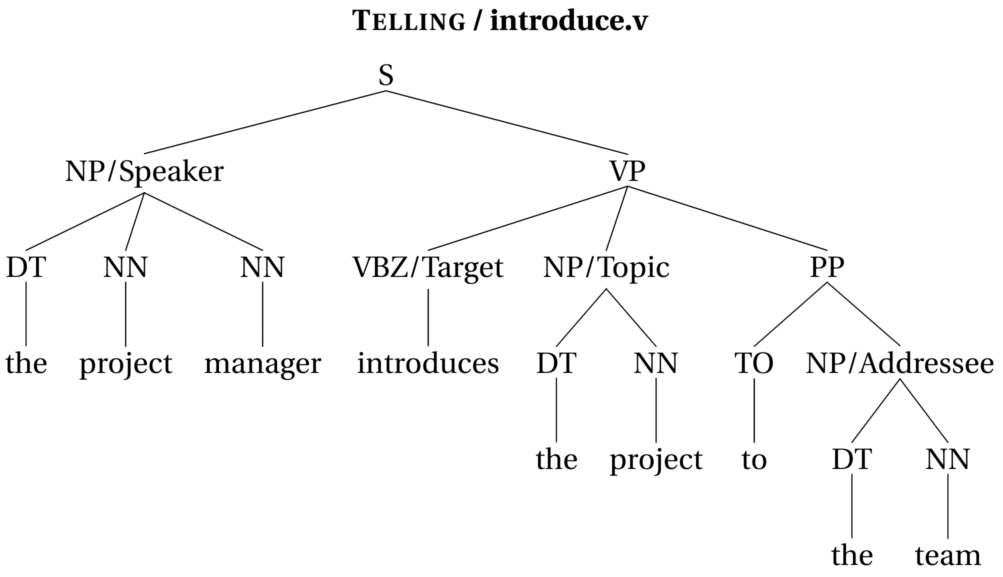


Figure 5.26: A saturated syntax tree after a lexicalization for each Frame Element has been chosen and added to the partial tree.

meeting to yield a concrete representation of its contents, transform its representation into a summary representation, and generate an English summary text from the latter.

This chapter discusses the representation of meeting contents for subsequent summarization, and introduces a general method how to arrive at such a representation given a recorded meeting.

For interpreting meetings, we acknowledge the usefulness of processing different modalities in addition to the conversational discourse, especially head and hand gestures. However, we argue that the overwhelming amount of information can be found in what the meeting participants discuss during a meeting. Therefore this chapter concentrates on methods for processing the spoken discourse. Starting from raw audio signals, this chapter discusses the involved sub-tasks for creating an automatic transcript of a meeting. This is in itself a difficult challenge, and not subject of this thesis. To avoid degradation by the influence of erroneous ASR transcripts, we use the manual meeting transcripts of the AMI corpus for all of our experiments. This is a voluntary restriction and not a flaw of the proposed approach. Rather, this design decision should be seen as an attempt to keep the experimental environment as controlled as possible, to avoid amplifying the mistakes of pre-processing steps that we can not influence. Once the theory of this thesis has matured, it would be an interesting experiment for future work to measure performance degradation when using automatic instead of manual transcripts.

The approach taken in this chapter to transform a meeting representation into a summary representation is motivated by evidence from cognitive science on human summarization strategies, expressed in the three macro-rules *deletion*, *abstraction*, and *construction*.

The macro-rules outlined in this chapter implement basic *reasoning* over propositions encoded as Semantic Frames. The objective is to reduce content. This means that some of the information contained in the source representation has to be deleted. The macro-rules accomplish that by either deleting propositions completely or in part.

For the first possibility, deleting a proposition entirely, we need a decision process that provides for every proposition whether or not it should be deleted or not, based on an idea of *relevance* with respect to the meeting summary. Relevance assessment has been widely studied for document summarization and also for speech-based scenarios. This chapter has presented the state-of-the-art findings in that area, which are typically based on statistical classification.

The approach proposed in this thesis draws on the results of [Murray, 2008] for the detection of deletable propositions. Since that work uses the textual representation of a meeting transcript directly, a procedure is discussed in this chapter that is able to adapt Murray's work to our own content representation with Semantic Frames.

For the second way to reduce information, namely by deleting only some parts of a proposition while keeping other parts, two strategies have been discussed. The first strategy abstracts away information either by dropping less relevant Frame Elements or by moving up to a more abstract Frame in the Frame hierarchy. The second strategy replaces sets of propositions with a .. Under the assumption that the new Frame describes a well-known and conventionalized situation, the consumer of a summary is expected to re-infer the original Frames using script knowledge.

In addition to content-reducing strategies, two further transformations were introduced that change the perspective on a given representation, either by changing a perspectivized Frame, or by changing the way entities are grouped.

Finally, this chapter also introduces a novel approach for generating English text from a framal content representation. A sentence is represented as a dependency tree of Frames, where the Frame Elements either refer to dependent Frames or to real-world entities.

The basic notion for generating Frame content is that of the partial syntax tree, a kind of template that maps the target and the Frame Elements of a certain Frame to syntactic phrase structures. Generating becomes the process of finding a suitable partial syntax tree from a library, and extending the free nodes until a full syntax tree is created.

For complex sentence structures, i.e., sentences that encode more than one Frame, an procedure to combine multiple partial syntax trees has been introduced.

A library of partial syntax trees can be generated automatically from annotated Frame data, as they exists in the FRAMENET corpus. This means, that syntactic-semantic rules do not have to be written by hand, but can be learned automatically.

To illustrate the approach, we also demonstrate the concrete operation of MEESU on parts of the transcript of the AMI meeting ES2008a.

Chapter 6

Evaluation

6.1 Introduction

This chapter concerns itself with the task of assessing the quality of a summary. We begin with a general outline of the difficulties inherent in that task, and introduce some common terminology (Section 6.2). Broadly speaking, evaluation is difficult because there is no absolute notion what makes a "good" summary. We discuss two general approaches to evaluation, *intrinsic* and *extrinsic* methods. The following three sections present the most widely used intrinsic metrics for assessing the quality of a generated meeting summary: ROUGE (Section 6.2), the Pyramid Method (6.2), and Basic Elements (Section 6.2). Finally, Section 6.3 introduces our own contribution to the task of summary evaluation, a specially designed experiment for extrinsic evaluation.

This *Decision Audit* experiment provides a framework with which different versions of summaries can be compared to each other or to reference summaries. The subjects in the experiments are asked to answer a certain complex information request using a set of recorded meetings with videos, transcripts, and summaries. Different subjects are provided with different summaries, allowing to assess the performance of the subjects across different variations of summaries.

We argue that our Decision Audit experiment is a valuable addition to the canon of existing evaluation methods because it measures the actual usefulness of a summary in a real-life task. In addition, the result of this extrinsic evaluation allows for a more fine-grained understanding of the quality of a summary because it compares various aspects of summaries, rather then expressing the quality of a summary with a single number.

6.2 Related Work

Summary evaluation is generally accepted to be a difficult task. It begins with having a clear understanding about *what exactly it is* that we are trying to evaluate here. Intuitive descriptions of an evaluation task such as "testing how well a summary performs with respect to the problem it addresses" fail because there are potentially many different problems for which a meeting summary could be beneficial. It is not at all obvious that one particular summary is suitable for all situations, it might very much depend on the practical needs of the summary user in a particular situation.

Also, it is not only the situation that might have an influence on what an ideal summary would look like; it is also depends on the person who wants to access the summary and what kind of prior knowledge he or she has. Among other aspects, this includes domain-dependent knowledge, i.e., what an (automatic) summarizer can savely assume to be generally known by its consumers about the domain of discourse. It also includes task-specific knowledge and general knowledge about how to use and work with summaries.

Independent from the user and his or her specific information need, we can also ask questions about the inherent quality of a summary *per se*, and here in particular about what Mani [2001] calls "informativeness" and "coherence" which, as he argues, are somewhat orthogonal dimensions. Informativeness measures how much of the information of the source, i.e. of a meeting, transpires in the summary. Coherence is a metric for the readability of a summary.

In the literature, these two general types of criteria are known as "extrinsic" and "intrinsic". We call methods extrinsic that test the usefulness of a summary *for a specific task* and methods which look only at qualities *inherent* in a generated summary intrinsic. Most of the intrinsic metrics widely used (see below) ignore qualities such as "coherence" and instead focus exclusively on criteria to measure "informativeness".

A standard approach in many NLP fields is to evaluate a system by comparing its overall performance with some kind of "gold standard", i.e., a reference that is considered to be (near-) optimal. Often times, such gold-standards are created manually specifically for evaluation purposes, but in some cases, such as, for instance, supervised machine-learning experiments, the gold-standard can be derived from other data, e.g., as a dedicated part of an annotated corpus.

Even manually produced gold-standard data is not necessarily error-free, which introduces a general problem for this kind of evaluation. If an automatic system produces a result that differs from the gold-standard it is usually assumed to be wrong. It is possible, though, that in fact the gold-standard is wrong in that particular case, falsely penalizing the system under evaluation. This raises the question of how to arrive at a good quality gold-standard that minimizes such errors.

One way to deal with this problem is to have not only one but multiple gold-standard references whenever the application domain is subject to opinion. This is a typical approach for summarization, where a gold-standard may consist of multiple hand-written summaries. It can be observed that summaries of the same document from different people can differ quite drastically. In one of the earliest studies of its kind, Rath et al. [1961] examine the reliability of human subjects in judging what a document is about. Six subjects are asked to select 20 sentences from each of ten scientific articles, so that these 20 sentences together form, in the subject's opinion, the most representative account of the article. It was observed that on average only 2.7 sentences per article were co-selected by all six candidates.

This might be taken as evidence that in general people differ in their assessment of what an article is about and thus which subset of an article's sentences should be chosen to represent the articles main contents. Interestingly though, the same subjects were asked to perform the same sentence selection task again two months after the original experiment. In this second round, each participant selected the same sentence as in their own first run in only 55 percent of the time on average. While it is possible that the participant's understanding what the articles were about had changed by the time of the second experiment, an alternative interpretation is that there is no single set of representative sentences for an article.

Rath et al.'s experiment allows for two observations. The fact that only 2.7 of sentences were co-selected by the six candidates can be explained either by the fact that different readers assess different parts of an article as important, or that the same extract-worthy information may occur in more than one sentence so that an extractor can freely choose between a number of more or less redundant sets of sentences. The former hypothesis is supported by the variety of opinions commonly found among different people, the latter one by the results of the second experiment two months after the first. However, it seems likely that the observed variation in the extractions result from a combination of both effects.

For gold-standard summaries in which the human summarizers basically annotate each sentence from the original document with a binary value (whether to include it in the summary or not), the inter-annotator agreement can be measured using Cohen's *kappa*. Although this measure has been criticized before [Banerjee et al., 1999], it is still widely used. For any annotation task in which a number of judges classify some units into mutually exclusive categories, Cohen defines the *kappa* measure as follows:

$$\kappa = \frac{p_0 - p_c}{1 - p_c}$$

where p_0 is the proportion of units in which the judges agree and p_c is the proportion of units for which agreement is expected by chance [Cohen, 1960]. The idea here is to correct the cases where annotations match by the probability that such an agreement occurred purely by chance. For extractive summarization experiments like Rath's, these units would typically be the sentences of the document, and p_0 and p_c could be estimated from relative frequencies.

Given a set of gold-standard summaries, the question is how these references can be used to measure the quality of an automatically produced summary. We are looking for a way to compare a candidate summary that was produced by an automatic system to the reference summaries in the gold standard. In the case of extractive summaries, it would of course be possible to use a metric like *kappa* to not only compute the agreement between gold-standard summaries, but in particular to compute the agreement between manual and automatically generated extracts. Alternatively, one could interpret finding the best sentences to extract as an information retrieval task and adapt traditional measures such as *precision*, *recall* and (weighted) *f-score* between a generated summary *S* and a gold-standard summary *GS*:

$$Precision = \frac{\#Sentences\ in\ both\ S\ and\ GS}{\#Sentences\ in\ S}$$

$$Recall = \frac{\#Sentences\ in\ both\ S\ and\ GS}{\#Sentences\ in\ GS}$$

$$F\text{-}Score = (1+\beta^2)\frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall}$$

However, for any document it can always be the case the relevant information that appears in a certain sentence may redundantly be expressed in another sentence. If *GS* only contains the first sentence while *S* only contains the second one, all of the above measures would unfairly penalize *S*.

The theory presented in this thesis is integrative in that it combines research results from various different areas into one coherent approach to meeting summarization. Therefore, it can be viewed–and potentially critized–from many different angles: the suitability of the used representation formalism, of the interpretation techniques, relevance assessments and the generation method must all withstand scientific and pragmatic demands as must the particular way in which we have integrated all these parts.

The full system and the components it consists of can and should be evaluated to get a clear understanding about what they can and what they cannot deliver. In a complex system such as the one presented here, it is worthwhile to not only evaluate the overall system performance by judging the generated summaries, but also to have a closer look at each of the involved components separately. This is, of course, quite a complex endeavor since each component requires a particular evaluation tailored explicitly to the role of the component in the overall system. We address this issue below.

However, the ultimate question to ask with respect to the integrated theory as a whole is that about the quality of the meeting summaries it generates. Therefore in this chapter, we are especially interested in an evaluation of the quality of the final system summaries. Not only does such an evaluation help us make reliable statements rather then guesses about the actual quality of our approach, it will also help us identify possible issues to improve in future work.

ROUGE

The variability among human judges poses an apparent problem for quality assessment, making it difficult to reliably judge the quality of a summary. Lin and Hovy take this as an argument to introduce a fully automated evaluation method. Such an approach makes evaluation less costly, more objective and repeatable. However, the authors also postulate that an automated metric should correlate positively with human judgment.

Inspired by previous work in machine translation, they modified a statistical metric by Papineni et al. [2002] for a summary evaluation method based on n-gram co-occurrence comparison with a set of gold-standard summaries. First conceived for the DUC document understanding conference, their method was further developed to a framework called "ROUGE" [Lin, 2004] which comes in five different variations:

ROUGE-N An *n-gram* is an n-tuple of words. If we understand a document S (article, meeting transcript, summary, etc.) as a sequence of k words, we obtain the set of n-grams contained in the document as all combinations of n words that appear in sequence in that document.

More formally, let $S = (w_0, \ldots, w_{k-1})$. The set of n-grams contained in S is defined as $G_n(S) = \{(w_j, \ldots, w_{j+n-1}) \mid 0 \leq j \leq k - n\}$.

For example, consider this sentence taken from AMI Meeting IS1009b: *"Is it yellow and black or is it yellow and blue?"* For $n = 3$ we obtain the following seven 3-grams (also called trigrams) from that sentence:

2× **(is, it, yellow)**	is it yellow and black or is it yellow and blue
2× **(it, yellow, and)**	is it yellow and black or is it yellow and blue
1× **(yellow, and, black)**	is it yellow and black or is it yellow and blue
1× **(and, black, or)**	is it yellow and black or is it yellow and blue
1× **(black, or, is)**	is it yellow and black or is it yellow and blue
1× **(or, is, it)**	is it yellow and black or is it yellow and blue
1× **(yellow, and, blue)**	is it yellow and black or is it yellow and blue

ROUGE-N is a recall-oriented measure that uses n-grams as the basic unit of comparison. Given a set of gold-standard summaries *GS*, we count how many times each n-gram, that occurs in the summary S to evaluate, also appears in each of the gold-standard summaries. This number is divided by the number of occurrences of the different n-grams in the gold-standard summaries, to yield a value between 0 and 1.

Let *Count(n-gram,X)* be the number of times *n-gram* appears in a summary X. Then we define:

$$ROUGE\text{-}N(S) = \frac{\sum\limits_{R \in GS} \sum\limits_{g \in G_N(R)} Count(g,S)}{\sum\limits_{R \in GS} \sum\limits_{g \in G_N(R)} Count(g,R)}$$

ROUGE-L ROUGE-N requires a fixed N to be defined prior to the evaluation, and it is not clear what is a good value for this N. The ROUGE-L variation relaxes this requirement somewhat. Instead of n-grams, ROUGE-L is based on the idea of a *Longest Common Subsequence (LCS).* A common subsequence of two sequences is a sequence that is a subsequence of both of the sequences. It is called a longest common subsequence if no other subsequence of the two sequences is longer.

Again we view a summary as a sequence of words, allowing us to compute in a first step the LCS for a pair of summary sentences between a candidate summary to evaluate and a reference summary. Thus a sentence in the reference summary becomes a sequence $R = r_1,...,r_m$ and a sentence in the candidate summary become a sequence $C = c_1,...,c_n$. Then we define $LCS(R,C)$ as the length of the longest common subsequence of C and R, and *sentence-level ROUGE-L* as a weighted f-score:

$$\begin{aligned} P_{lcs}(R,C) &= \frac{LCS(R,C)}{m} \\ R_{lcs}(R,C) &= \frac{LCS(R,C)}{n} \\ F_{lcs}(R,C) &= \frac{(1+\beta^2)R_{lcs}(R,C)P_{lcs}(R,C)}{R_{lcs}(R,C)+\beta^2 P_{lcs}(R,C)} \end{aligned}$$

With these definitions, Lin defines *summary-level ROUGE-L* as the union of sentence-level ROUGE-L between a candidate summary C and a reference summary R. Let us assume that C is a sequence of words $c_1,...,c_n$ and R consists of K sentences $R_1,...R_K$ with a total number of m words. Then we get:

$$\begin{aligned} P_{lcs}(R,C) &= \frac{\sum_1^K LCS(R_K,C)}{n} \\ R_{lcs}(R,C) &= \frac{\sum_1^K LCS(R_K,C)}{m} \\ F_{lcs}(R,C) &= \frac{(1+\beta^2)R_{lcs}(R,C)P_{lcs}(R,C)}{R_{lcs}(R,C)+\beta^2 P_{lcs}(R,C)} \end{aligned}$$

ROUGE-L again is defined as the weighted f-score for a non-negative β.

ROUGE-W ROUGE-L computes the same value for any subsequence with maximal length, ignoring the distribution of the words in the sequence. In some situations it might be desired to give those subsequences a higher score that contain longer *consecutive* matches. Therefore ROUGE-W, defined through a dynamic program given in [Lin, 2004], rewards those subsequences higher than subsequences in which the words are more spread out.

ROUGE-S A *skip bigram* is a pair of words from a sentence in their sentence order. For a sentence with k words, we thus get $C(k) = \frac{k(k-1)}{2}$ different skip bigrams[1]. We can then define recall and precision in terms of skip bigram matches between two summaries as follows. Let SKIP2(R,C) be the number of skip bigram matches between candidate summary C and reference summary R, and $\beta > 0$.

$$\begin{aligned} P_{skip2}(R,C) &= \frac{SKIP2(R,C)}{m} \\ R_{skip2}(R,C) &= \frac{SKIP2(R,C)}{n} \\ F_{skip2}(R,C) &= \frac{(1+\beta^2)R_{skip2}(R,C)P_{skip2}(R,C)}{R_{skip2}(R,C)+\beta^2 P_{skip2}(R,C)} \end{aligned}$$

Again, we define the actual ROUGE metric as the weighted f-score of recall and precision.

As a variation, ROUGE-S allows to define a maximal skip distance d_{skip} between the two words of a skip bigram, i.e., only those skip bigrams are considered where the distance between the components if at most d_{skip} words. This variation is then called ROUGE-Sd_{skip}, e.g. for $d_{skip} = 4$ we get ROUGE-S4.

ROUGE-SU Even though two summaries share a decent amount of words, their ROUGE-S scores may turn out low if the order in which the common words appear in the sentences are different enough not to produce many common skip bigrams. Therefore, ROUGE-SU is a variation of ROUGE-S that also counts unigram matches between candidate and reference.

Lin [2004] also estimates the correlation between the different ROUGE metrics and human judgments on the basis of data from the DUC 2001, 2002, and 2003 conferences.

[1] This is the number of skip bigrams with respect to the word positions in the sentence. Some of these skip bigrams may consist of the same words as others, though, when the sentence contains some words multiple times.

The Pyramid Method

Methods based on comparing the surface form of summaries with that of gold standard summaries have a naturally favor extractive summarization methods. Since extractive summaries consist of material taken verbatim from the original source document, the sentences in the summary will be identical to some of the sentences from the document and the same holds for the sentences of the reference summaries. In the unexpected case, in which the system to evaluate extracts completely irrelevant sentences, in the sense that they do not contain summary-worthy content, those sentences will have no or only a very small overlap with the sentences in the gold standard summaries. If, however, the automatic summaries does extract informative material, the generated summaries will with a high probability contain the exact same sentences and/or phrases as the reference summaries since both extract from the same source. The final score becomes a matter of weighting and summing up the matching parts.

For abstractive summaries however, which are generated from scratch, it is not even clear whether they will display any textual overlap in the surface form at all, *even when they contain highly relevant content.* In a generated abstract, the choice of wordings is taken by the generator component and typically independent of the wordings in the source document. For instance, the use of synonyms, a different voice, different grammatical person, less colloquial wordings, etc. will all have a big effect on the surface realization. Measuring how many, say, n-grams appear both in the generated summary and a reference can therefore not be considered a fair metric, it is very likely to yield low results for newly generated yet high quality content.

Nenkova and Passonneau [2004] take the problems of matching mere surface forms as an argument for their “Pyramid Method” which is related to a similar idea by van Halteren and Teufel [2003]. What both approaches have in common is the use of a more content-oriented representation of summaries. They collect atomic meaning units in summary sentences, called “summary content units” (SCU’s) by Nenkova and Passoneau and “factoids” by van Halteren and Teufel. While the factoids approach is used to study consensus between multiple human summaries in order to estimate the degree of variation among a set of summaries and whether they are generally suited for instrinsic evaluation, the Pyramid Method extends this principal idea and proposes a specific evaluation method for summaries based on it.

Starting point for the Pyramid Method is a set of gold-standard summaries. SCU’s are derived through an annotation procedure in which sentences of similar content are identified in the summaries. In these sentences, the basic facts shared by the different sentences are identified as distinct SCU’s. In a similar practice, also those SCU’s that only appear in a single summary are recorded. In the resulting set of SCU’s, each entry consists of:

- a unique index
- a weight
- a natural language content description
- a list of contributors

The weight of an SCU is simply the number of gold-standard summaries it appears in. The list of contributors links summaries and SCU's together: for each summary that contains a particular SCU those words are identified in the summary that together evoke the meaning of the SCU's. These words are called the "contributor". Each word in a summary has to be part in exactly one contributor which in turn has to contribute to exactly one SCU.

The method then proceeds to sort the SCU's into vertical tiers, with ascending weight from bottom to top. Since they assume that the number of SCU's shared by most or all of the summaries will be quite low while the number of SCU's only found in single summaries will be rather high, a visualization of these tears would then resemble the pyramid that gives the method its name (see. Figure 6.1).

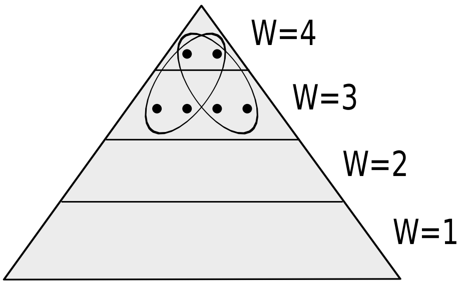


Figure 6.1: A pyramid of six SCU's, two of which are shared by four summaries and four of which are shared by three summaries each (taken from [Nenkova and Passonneau, 2004]).

The pyramid allows to derive a "consensus" summary, i.e. a summary that represents a mean of all given summaries, by successively including SCU's from the highest tier downward until the desired summary length is reached. Nenkova and Passonneau consider all SCU's in one tier to be equally important and thus do not specify a selection method within a tier. Therefore, multiple consensus summaries with equal weight are possible if the lowest tier that contributes to the summary contains more than one SCU.

With the Pyramid Method, it is now possible to compute a score for each summary. First, the total weight of a summary is computed as the sum of all the SCU's it contains. Then the weight of a consensus summary with the same number of SCU's is computed analogously. The final score is the ratio between the two numbers, ranging from 0 to 1.

Van Halteren and Teufel did various experiments based on such a consensus summary over a set of 50 gold-standard summaries of a BBC report. It is interesting to note that in their experiments, only one factoid (roughly equivalent to a SCU) was shared by all of the summaries. The content of this factoid could be verbalized in a simple sentence of only three words[2]. In order to create a consensus summary of 100 words–with 100 being the target length of the original set of gold standard summaries–the authors had to move quite low in the pyramid: the last tier used for the summaries contained factoids that occurred in at least 30% of the summaries.

Basic Elements

One of the disadvantages of the Pyramid Method over a method like ROUGE is that it cannot be run fully automatically because the identification of the SCU's in the reference summaries as well as in the summary to be evaluated requires manual interaction. To overcome this issue, Hovy et al. [2006] pick up the basic ideas of the Pyramid Method and introduce a framework which takes a general, abstracted view on methods based on content units, but which at the same time can be implemented through automatic processes. Their method consists of three preparatory steps and three scoring steps:

1. Extract from the gold-standard summaries a set of content units, called *basic elements* (BE's).

2. Identify similar BE's and match them together.

3. Score and rank the list of reference BE's.

4. Identify occurrences of BE's in the summary to be evaluated.

5. Match the extracted BE's against the reference list of BE's

6. Integrate the scores of the matched BE's into a final score.

Functionally, the method requires four distinct modules, a *BE-Breaker* to extract BE's from a summary, a *BE-Matcher* to identify identical BE's, a *BE-Scorer* to assign an individual score to each BE, and a *BE-Score Integrator* to compute the final score of a summary from the scores of the BE's it contains.

In a first implementation of this approach which the authors use to evaluate data from the document understanding conference DUC-2005 [Hovy et al., 2005],

[2] The original article which the gold standard summaries were based on, was about the backgrounds of the murder of the Dutch politician Pim Furtuyn. The single factoid shared among all summaries stated “Fortuyn was murdered”.

the following definition for basic elements was used: a BE is either the head of a major syntactic constituent (noun, verb, adjective or adverbial phrases) or a triple `(head | modifier | relation)` where `head` is again the head of a syntactic constituent and `modifier` is a single dependent. The relation may be empty or express the specific semantic relationship between the head and the modifier.

For instance, one incarnation of Hovy et al.'s method produces the following BE's for the sentence *"Two Libyans were indicted for the Lockerbie bombing in 1991"*:

```
<Libyans | two | CARDINAL>
<indicted | Libyans | ACCUSED>
<indicted | bombing | CRIME>
<indicted | 1991 | TIME>
```

To arrive at such BE representations, syntactic parse trees have to be generated automatically. Four different parsers were used together with specialized "cutting rules" to extract BE's from the resulting parse tree. Every BE from the candidate summary is matched against all BE's from the gold-standard summaries. It receives one point for every gold-standard summary it matches. The authors outline a number of different strategies how to compute a match between two BE's, two of which are available in their implementation: *lexical identity* and *lemma identity*. Lexical identity matches two BE's together if the first two components of the BE's are exactly the same; lemma identity matches two BE's when the canonical word form (lemma) of their components are the same. The user can further choose whether or not the relation component has to match, too (condition *HMR*) or whether it will be omitted (condition *HM*). The overall score for a candidate summary is computed as the sum of the scores of all of its BE's.

The authors compared their system output with the average scaled responsive score for each summarizer as computed by NIST, in order to assess the correlation between the BE metric and human judgment. The BE results correlated quite well with these values, with a Spearman coefficient of about 0.928 and a Pearson coefficient of about 0.976. In addition, they provided correlation values with ROUGE and Pyramid scores, shown in Figure 6.2.

A second implementation called "BEwT-E" uses a slightly different notion of a basic element [Tratz and Hovy, 2008]. Here, a basic element is a list of one to three words together with their part-of-speech tag. The system utilizes a number of state-of-the-art NLP tools as external modules, among them a named-entity recognizer (NER). For entities recognized with that tool, the part-of-speech tag may be replaced with the type of the named entity.

The BE's consisting of only one word are all the nouns, verbs and adjectives found in the summary. For the two-word BE's, a number of typical syntactical constructions are allowed, such as, `(subject, verb)` or `(adjective, noun)` etc. The components of such pairs are taken from the single-word BE's. Triple word BE's

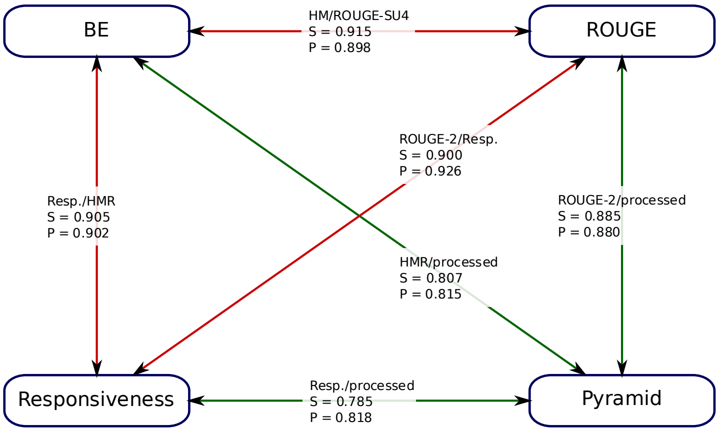


Figure 6.2: Correlation between different intrinsic evaluation metrics (reproduced from Hovy et al. [2005])

consists of two single-word BE's connected via a preposition or a functional relation, such as, "because", "where", etc. The following list exemplifies the different variations:

- **Unigram BE:** `(milk:NN)`
- **Bigram BE:** `(green:JJ, plant:NN)`
- **Trigram BE:** `(rejection:NN, of:IN, John:NNP)`
- **NE instead of POS tag:** `(rejection:NN, of:IN, John:Person)`

In order to arrive at these basic elements, the authors combine a syntax parser [Charniak and Johnson, 2005], the LingPipe NE recognizer[3] and regular expressions over parse trees [Levy and Andrew, 2006]. Each BE is scored either according to the numbers of reference summaries it appears in, the square root of that value or a constant weight of 1.

The systems biggest difference to the original version is, however, the complex matching mechanism between two BE's. Instead of pure lexical matching between the components, a list of variations is created automatically from each BE by applying certain transformation operations. These are:

(De-)lemmatization Instead of the original word form, the word's lemma is used.

Synonymy The word is replaced with another word with equivalent meaning. Strict synonym replacement relies on a word's most frequent WordNet [Fellbaum,

[3] http://alias-i.com/lingpipe/

1998] sense, but variations are the use of abbreviations, exchange of similar prepositions [Litkowski and Hargraves, 2005, cf.], and length variations of proper names.

Generalization Nouns standing for specific instances of a class can be replaced with a nound representing the class. Named entities may likewise be repressed by the nouns expressing the named-entity type.

Complex noun constructions complex nominal construction might alternatively be expressed as verb-noun or adjective-noun constructions or vice versa. Noun pairs may be reversed or expressed with a possessive of pronoun.

Pronouns Pronouns may match names and third person plural pronouns are allowed to match companies.

For the scoring part, BE's from the reference summaries are extracted and collected in a reference pool. Depending on the setup, duplicates may optionally be removed from this pool automatically. The summary to be evaluated is parsed and BE's are extracted. Each original BE is then matched against the pool together will all possible transformations from the above list. Because a BE and its transformation may result in several positive matches, and since the BE's in the pool may have different weights, the computation of the final match score is treated as a weighted assignment problem. For multiple reference summaries, a jackknifing procedure is implemented to allow a fair ranking in case the gold-standard summaries are included in the rank themselves.

The authors noted higher scores in an evaluation performed a machine translation scenario when the above transformations were used. Also, Pearson and Spearman coefficients with manual adequacy judgments were significantly higher with transformations. Unfortunately, no results for a summarization task are given.

While the BEwT-E method contains an impressive list of transformations which significantly boost recognition of similar BE's and correlation scores, this method is highly dependent on external software modules. Most of these modules do not achieve 100% accuracy which may affect results for different summaries. For instance, a correct summary may be penalized only because the used syntax parser misparses the contained sentences. Also, it is unclear how this implementation will handle updated versions of the external modules as future versions of the used syntax parser or the actively developed WordNet will likely result in different scores for an otherwise identical experiment. This makes comparability and repeatability difficult.

Other than that and the fact that the authors are still experimenting with a final parameter setup, the method is a promising approach that brings together results from different NLP areas to an integrated application.

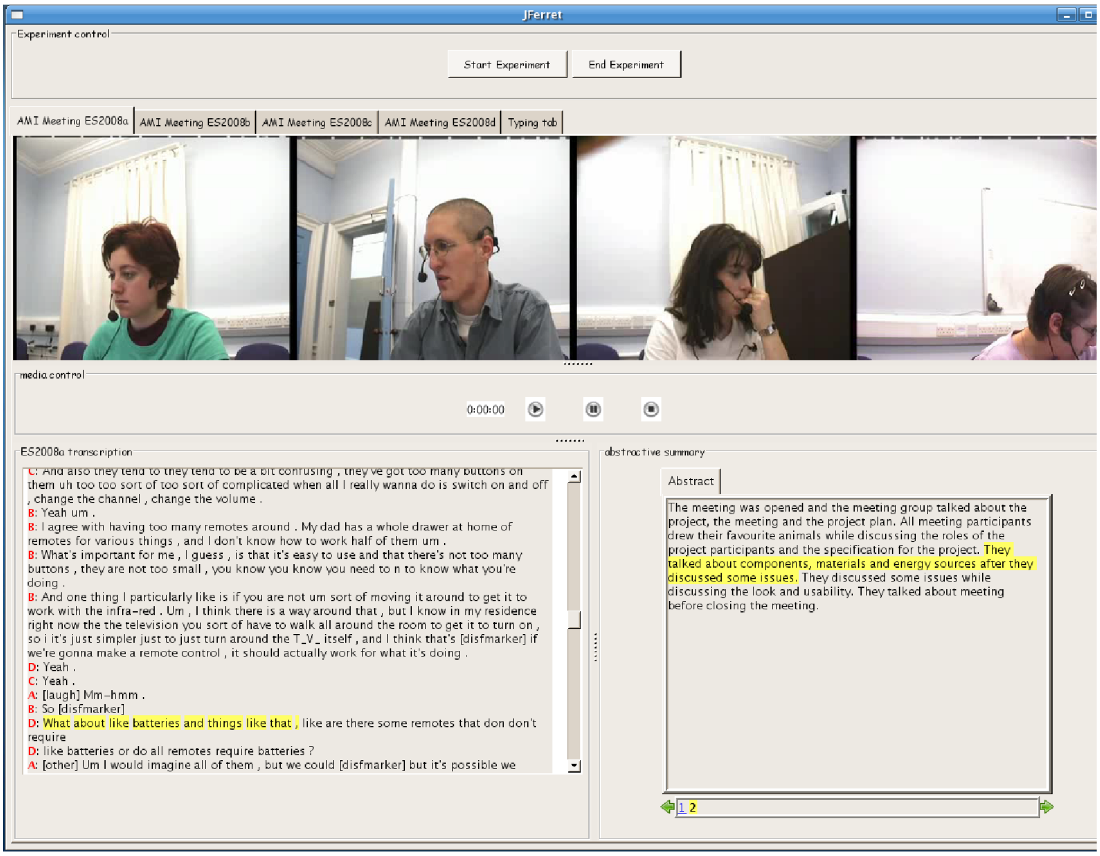
6.3 Extrinsic evaluations

To assess the quality of different kinds of meeting summaries extrinsically, a dedicated evaluation framework called “Decision Audit Task” has been designed as part of this thesis, in collaboration with Gabriel Murray (University of Edinburgh) and co-workers from DFKI [Murray et al., 2009]. The original experiment compares three different meetings summaries with a baseline and topline (gold-standard) condition. 50 subjects took part in the experiment with 10 subjects per condition. The task in each condition was identical and only the condition setup was changed to reflect the five variations.

For the experiment, the subjects are presented with a special meeting browser that displays a set of four meetings from the AMI corpus. We chose a set in which the meeting participant take the design of a new remote control serious and collaborated well. They show a careful and deliberate decision making process in every meeting and across the series as a whole. Particular attention was paid to these points because of the main task in the “decision audit” experiment: here, the subjects are asked to write a short paragraph summarizing the decision making process in the four meetings about the separation of often and rarely used function of the remote control.

Figure 6.3 shows a screenshot of the meeting browser for one of the five conditions, the *topline* condition. It consists of four major “tabs”, one for each of the four meetings plus one for the subjects’ written answer. Every tab shows a close-up video of each of the four participants from the particular meeting. The videos were synchronized with each other and with a down-mix of the audio recodings of the meeting. The subject can freely control the playback of the audio/video through the start, pause, stop-controls under the video. Each condition displays a transcript of the meeting in the lower left half of the meeting browser. Clicking on a particular point in the transcript will take the audio/video streams to that point in the meeting. The difference in the setup of the five tested conditions lies in what was displayed in the lower right corner of the meeting browser. The baseline condition is a simple keyword-based condition. Here, the subjects are given a hyperlinked index of the 20 keywords with the highest `SU.IDF` values [Murray and Renals, 2007], a variant of `TF.IDF`. Clicking on a keyword opens a small list with one entry for each point in the meeting transcript where the keyword occurs. These entries are hyperlinked and clicking on them will take the user to their point of reference in the transcript. In addition to displaying this entry list, clicking on a keyword will make the transcript automatically jump to the point of the first entry in the list.

The other four conditions, including the topline, are all summary conditions. They differ in the method through which they were created and in the data that was used in their respective creation process. All summaries are all based on transcripts of the meeting they summarize, where one of the conditions, Extractive2, uses an ASR transcript while all the others used a manual transcript. Two of the



C: And also they tend to they tend to be a bit confusing , they've got too many buttons on them uh too too sort of too sort of complicated when all I really wanna do is switch on and off , change the channel , change the volume .
B: Yeah um .
B: I agree with having too many remotes around . My dad has a whole drawer at home of remotes for various things , and I don't know how to work half of them um .
B: What's important for me , I guess , is that it's easy to use and that there's not too many buttons , they are not too small , you know you know you need to n to know what you're doing .
B: And one thing I particularly like is if you are not um sort of moving it around to get it to work with the infra-red . Um , I think there is a way around that , but I know in my residence right now the the television you sort of have to walk all around the room to get it to turn on , so i it's just simpler just to just turn around the T_V_ itself , and I think that's [disfmarker] if we're gonna make a remote control , it should actually work for what it's doing .
D: Yeah .
C: Yeah .
A: [laugh] Mm-hmm .
B: So [disfmarker]
D: What about like batteries and things like that , like are there some remotes that don don't require
D: like batteries or do all remotes require batteries ?
A: [other] Um I would imagine all of them , but we could [disfmarker] but it's possible we

The meeting was opened and the meeting group talked about the project, the meeting and the project plan. All meeting participants drew their favourite animals while discussing the roles of the project participants and the specification for the project. They talked about components, materials and energy sources after they discussed some issues. They discussed some issues while discussing the look and usability. They talked about meeting before closing the meeting.

Figure 6.3: A screenshot of the meeting browser used in one condition of the Decision Audit Task

conditions use extractive summarization techniques [Murray, 2008]. One uses a semi-automatic abstractive approach [Kleinbauer et al., 2007b], a predecessor to the approach presented in this thesis. Although the core summarization algorithm is fully automatic, the particular instance in the experiment uses manually annotated dialog act and topic segments. The topline condition is a manual abstract based on manual transcript. Table 6.1 gives an overview of the conditions.

The experiment is time-constrained. After an intial familiarization phase with the browser using an unrelated set of meetings, a subject has only 45 minutes to finish the complete task, including writing the answer summary. The four meetings used in the actual experiment last circa 17, 37, 35, and 44 minutes respectively and in that light, it is a challenging task. In particular, it means that the subjects do not have enough time to simply play back the four meetings and then write the answer. This was a deliberate design decision for the experiment, in order to "force" the subjects to use the browsing facilities provided by their condition.

After the experiment is over, a post-task questionnaires has to be filled out by

Table 6.1: The different conditions of the Decision Audit Task experiment.

Condition	**Description**
Baseline	Top 20 keywords from transcript
Extractive1	Extractive summary on manual transcripts
Extractive2	Extractive summary on automatic transcripts
Abstractive	Abstractive summary on manual transcripts
Topline	Hand-written abstracts

each subject. Here, they are presented ten statements and they indicate their own level of agreement or disagreement with each statement on a 5-point Likert scale. The average scores assigned to the statements by the 50 subjects are shown in Figure 6.4. As the results indicate, the subjects found the experiment challenging.

In addition to the questionnaires, the usage of the meeting browser by a subject is analyzed through low-level events logged from the graphical user interface. In this log, key- and mouse events were recorded with the exact timings of their occurrence. This allows to analyze the browsing behaviour of a participant and one goal of the study was to find out whether there are significant differences across the five conditions.

As a third evaluation track, the quality of the answers of the subjects is assessed in two different ways, a somewhat more *subjective* measure and a more *objective* measure.

For the *subjective* evaluation, two independent judges first read through all 50 answers to get a first impression of the variety of answers. They then rate six different criteria on 8-point Likert scales. These criteria are: *overall quality*, *conciseness*, *completeness*, *task comprehension*, and *participant effort*. Table 6.2 shows the results of this part of the evaluation. It is apparent that the manually written summaries that served as the gold-standard condition outperformed all of the other conditions. However, in most of the criteria, the generated abstracts were runner-ups to the topline. For criterion "writing style", they even scored slightly higher. The extractive conditions, especially on manual transcripts, didn't result in a huge gap with respect to their usefulness, they clearly outperformed the baseline condition.

The *objective* measure is based on the idea of deriving a "consensus answer" based on all 50 answers. A "gold-standard" list consisting of 25 items that an ideal answer should contain was created by three judges. Two of these judges then proceed to independently check all 50 subject answers against the item list. Then they meet again to compare the individual checking results. In 12 out of the 50 cases, the ratings of the two judges diverged by more than two items. For these cases, the

Question	Baseline	Extractive1	Extractive2	Abstractive	Topline
Q1 *I found the meeting browser intuitive and easy to use*	3.8	4.0	3.0	3.7	4.3
Q2 *I was able to find all of the information I needed*	2.9	3.8	2.9	3.0	4.1
Q3 *I was able to efficiently find the relevant information*	2.8	3.4	2.5	2.7	4.0
Q4 *I feel that I completed the task in its entirety*	2.3	3.1	2.3	2.9	3.2
Q5 *I understood the overall content of the meeting discussion*	3.8	4.5	3.9	3.9	4.1
Q6 *The task required a great deal of effort*	3.0	2.6	3.9	3.2	3.1
Q7 *I had to work under pressure*	3.3	2.6	3.3	3.1	2.7
Q8 *I had the tools necessary to complete the task efficiently*	3.1	4.3	3.0	3.5	4.1
Q9 *I would have liked additional information about the meetings*	3.0	2.0	2.4	2.7	2.6
Q10 *It was difficult to understand the content of the meetings using this browser*	2.1	1.5	2.7	2.3	2.0

Figure 6.4: Post-task questionnaire with average results

judges consult the underlying answers again and reach an agreement so that a final objective score can be computed. Of a theoretical maximum of 25 correctly identified items, the average results for the objective evaluation turned out quite low:

Baseline	Extractive1	Extractive2	Abstractive	Topline
4.25	7.2	5.05	7.4	9.45

Nevertheless, the abstractive condition scored second-best again, after the topline.

A final evaluation step analyzes the browsing behavior of the subjects in greater detail by means of the low-level GUI events recorded in the log file. A number of measurements were taken from these logged events, as depicted in table 6.3. One

Table 6.2: The results of the subjective evaluation

Criterion	**Baseline**	**Extractive1**	**Extractive2**	**Abstractive**	**Topline**
Overall quality	3.0	4.2	3.1	4.3	4.7
Conciseness	2.85	4.3	3.1	4.5	4.9
Completeness	2.55	3.6	2.6	3.9	4.5
Task comprehension	3.25	5.2	3.7	4.7	5.3
Participant effort	4.4	5.2	3.7	4.9	5.3
Writing style	4.75	5.7	4.1	5.8	5.7

Table 6.3: Results of the log file evaluation

Measurement	**Baseline**	**Extractive1**	**Extractive2**	**Abstractive**	**Topline**
Task duration	45.4	43.1	45.4	43.2	45.42
Beginning of first typing	16.25	13.9	17.14	10.22	8.61
Average tab switches per minute	0.98	0.81	0.72	1.13	1.4
Average summary clicks per minute	0.39	0.11	0.08	0.18	0.08
Average clicks per minute	1.33	2.24	1.47	0.83	1.99
Clicks on media controls	15.4	14.4	40.4	20.6	16.6
Correlation between clicks and writing	0.03	0.01	0.01	0.01	0.01
Length of unedited answer	1400	1602	1397	1650	2043
Length of edited answer	1251	1384	1161	1430	1760
Number of meetings viewed	3.9	4.0	3.9	4.0	4.0
Average writing timestamp	0.68	0.73	0.76	0.65	0.65

observation to make from this analysis is that extractive summaries on ASR transcript result in users using the recorded media much more often. Another finding is that in all conditions the users needed the full 45 minutes to finish the task, underlining the challenging nature of the experiment.

The original experiment used a previous version of our work that relied in parts on a manual analysis of the content structures of a meeting. We thus reran the Decision Audit task for the MEESU system with five subjects, three male and two female. Only the subjective and the objective evaluations and the questionnaires were considered for evaluation, the log file evaluation was not. The results are shown in Figure 6.5.

One should note that compared with the original experiment two different judges performed the subjective and the objective evaluation. The results that MEESU achieves often come close even to the manual topline, especially in the subjective evaluation. However, especially for it is unclear in how far the absolute scores are really comparable to those of the original experiment–after all, they are *subjective* and depend on the actual judges.

In comparison, the objective score that MEESU achieves is rather low. A possible explanation for this is that of the five subjects of this experiment, one person displayed a low motivation to work on the task thoroughly. While most other subjects pointed out the restrictive time limit of the task, that person was the only one who finished after only 30 minutes. The provided answer scored well below that of the other subjects and had a strong negative influence on the computed averages, especially in some of the questions in the post-questionnaire.

6.4 Chapter Summary and Discussion

This chapter gives an overview over standard evaluation metrics used in the field of summarization. However, we argue that such *intrinsic* measures fail to capture the usefulness of summaries in a concrete application. To address this issue we introduce a novel *extrinsic* evaluation framework, the Decision Audit Task. Subjects in this evaluation are presented a meeting browser that gives them access to recorded meetings, including a transcript and the summary condition to evaluate. The task is deliberately designed to have a very strict time limitation to encourage the subjects to make use of the summary functionality. The question involves a complex information need and cannot be answered in a single sentence.

The Decision Audit Task represents the largest extrinsic evaluation of meeting summarization to date. One of the conclusions that can be drawn from this study is that the participants considered automatically generated summaries to be coherent and useful. They generally outperforming a baseline of hyperlinked keywords in the subjective and objective evaluation criteria.

Question	Baseline	MEESU	Topline
Q1 *I found the meeting browser intuitive and easy to use*	3.8	3.8	4.3
Q2 *I was able to find all of the information I needed*	2.9	3.8	4.1
Q3 *I was able to efficiently find the relevant information*	2.8	3.0	4.0
Q4 *I feel that I completed the task in its entirety*	2.3	2.2	3.2
Q5 *I understood the overall content of the meeting discussion*	3.8	4.2	4.1
Q6 *The task required a great deal of effort*	3.0	3.4	3.1
Q7 *I had to work under pressure*	3.3	3.6	2.7
Q8 *I had the tools necessary to complete the task efficiently*	3.1	3.6	4.1
Q9 *I would have liked additional information about the meetings*	3.0	3.2	2.6
Q10 *It was difficult to understand the content of the meetings using this browser*	2.1	1.8	2.0

Criterion	Baseline	MEESU	Topline
Overall quality	3.0	4.7	4.7
Conciseness	2.85	4.8	4.9
Completeness	2.55	4.4	4.5
Task comprehension	3.25	5.1	5.3
Participant effort	4.4	4.7	5.3
Writing style	4.75	5.3	5.7
Objective score	4.25	6.6	9.45

Figure 6.5: The results of the Decision Audit evaluation for MEESU

While extractive summaries might be quite different in appearance from the way people write summaries, the Decision Audit Task shows that they can indeed be useful tools in the context of a meeting browser. However, the comparison with the hand-written gold-standard summaries shows a substantial gap in the performance of the participants. This is encouraging for future research on abstractive meeting summarization.

In the gold-standard as well as in the semi-automatic abstractive condition, the participants began writing their answers earlier and authoring more comprehensive answers. This suggests that abstractive summaries help participants to better and more quickly understand the gist of a meeting.

Chapter 7

Summary and Conclusions

This thesis describes a novel approach to abstractive meeting summarization. We have shown that this is a challenging task which has not received much attention in previous research, despite its doubtless usefulness in everyday application. In the introduction chapter, we identified a number of concrete research questions which we have addressed in this work. In the following, we summarize our main results with respect to these questions.

What is a viable design for an abstractive meeting summarization system?

We presented the MEESU system in Chapter 5 which implements a pipeline architecture to generate an abstractive textual summary from a previously recorded meeting. First, a transcript of the meeting is generated either automatically or manually(see Section 5.3). The transcript allows the contents of the discourse to be transferred into a lexicalized meeting representation 5.4 which is transformed into a lexicalized representation of the summary. The final step of the MEESU system is the generation of a textual abstract from the summary representation.

Can such a design be implemented using readily available knowledge sources for language processing?

A key question for abstractive approaches is how to represent both source and summary contents. Previous approaches have usually opted for specially crafted domain models. We have presented an approach that uses *Frame Semantics* at the heart of its representation formalism (see Sections 5.4). The main rationale for this choice is to use an existing formalism that is not tailored to a specific domain but designed for generality. At the same time the chosen formalism must lend itself to the task. A pragmatic demonstration that this lexicalized representation is suitable for the task of automatic meeting summarization is the design and the implementation of a working prototype.

How can a meeting transcript be transferred into such a representation?

A second advantage that arises from the usage of Frame Semantics is its natural proximity to the natural language input the system faces in form of meeting transcripts. Being a theory of lexical semantics, Frame Semantics is intrinsically designed to model natural language. That means that for the interpretation of a recorded meeting, we can rely on existing Frame parsers that have been trained on corpora such as the FRAMENET corpus. More specifically, our implementation uses the SE-

MAFOR parser which has been trained on the Wall Street Journal corpus, demonstrating the relative domain independence of the chosen approach. A shortcoming is, however, the limited coverage of FRAMENET, and thus implicitly of the parser. Although FRAMENET defines over a thousand unique Frames, the parser cannot always produce the correct Frame analyses for a given utterance. However, as FRAMENET is an ongoing research effort, existing gaps in coverage are expected to decrease.

Can insights from cognitive science be leveraged as constraints for deriving the contents of the meeting abstract?

A third new aspect is the adoption of insights of cognitive science into an actual computational implementation. Manually produced summaries are still unmatched by automatic procedures today, thus it is reasonable to study the ways in which human summarizers reduce source contents to meeting contents, and apply these findings in automated systems as well. The thesis at hand does that by implementing the *macro-rules* described by van Dijk and Kintsch [1983] in Section 5.5. But we also introduce two additional information retaining rules–*Change of Perspective* and *Grouping of Entities*–which are not themselves intended for content reduction, but for enabling macro-rules. They achieve this by transforming parts of the representations into semantically comparable representations that–unlike the original–fit the preconditions of the macro-rules. Without this contribution, the *transformation* phase from meeting to summary representation could not unfold its full potential, since the invocation of macro-rules would not be possible in some cases.

How can a content representation based on Frame Semantics be verbalized as text?

Abstractive approaches to summarization are rare in general, and the existing ones often times concentrate on content representation and/or source interpretation. The generation of an actual textual summary is not pursued in such cases. The approach at hand embraces text generation. Section 5.6 proposes a novel way to verbalize what is represented by a Semantic Frame: to the best of our knowledge, this is the first general purpose natural language generator for Frame Semantics.

Given a set of specific Frames, target lexical unit and Frame Elements, the question for text generation is how to map this semantic representation to syntactic and finally surface forms. In our approach this is done using *partial syntax trees* which are derived automatically from corpus annotation in a pre-processing step. This has to be done only once and is then used as a knowledge base within the generator. Again, the versatility of the Frame Semantic approach is underlined because no additional annotation is required. The existing annotation in the FRAMENET corpus can be re-used.

How can the usefulness of a meeting summary be measured?

Finally, we present a new methodology to evaluate the generated meeting summaries (Section 6). While the evaluation of automatic summarization is generally acknowledged to be a difficult task, a certain set of metrics, such as ROUGE, have become quasi-standards in recent years. Such *intrinsic* metrics try to measure an inherent quality of a given summary by comparing it with a set of gold-standard summaries. Ultimately, however, what decides about the utility of a summary is how well it supports its user in performing a certain task.

We therefore propose a new framework for a specific *extrinsic* task, a "decision audit". In this framework, users are presented a meeting browser that allows them to access a set of recorded meetings in the form of audio and video recordings, transcript and a summary for each meeting. Their task is to audit the decision making process of the meeting participants over the course of all meetings. Their performance is measured in a number of different categories and compared to other summary systems or to a baseline system.

7.1 Future Work

A thesis on a previously rather neglected sub-field of summarization almost certainly means that there is room left for improvement. In future work, we plan to extend the so far purely language centered approach to include multimodal information. A number of research projects with different foci have demonstrated how a automatic system can benefit from combining multiple modalities. For meetings, it is straight-forward to see that such information as nodding or shaking one's head, which are often used modes of conversation, is something that escapes a treatment of meeting contents based exclusively on the transcript.

But even for the modalities and information layers that were used, moving from the controlled, high-quality manual versions to automatically produced versions, is a future task. This includes the use of ASR transcripts, or automatically recognized dialog acts. Such a change is likely to reduce the quality of downstream processing, for instance, the recognition of Semantic Frames in the transcript.

On the other hand, tools such as SEMAFOR which is used for the automatic Frame prediction in this thesis typically treat their input utterances in isolation from the context they appear in. We suspect that especially for the interpretation of conversational discourse, a context model could have a beneficial effect on Frame parsing. This point is directly related to the fact the FRAMENET despite continually growing in size has still room for improvement in terms of coverage.

On the generation side, intelligent methods to backtrack from generation failure could be added. Such strategies would allow an intelligent replanning of the document structure if the generation of a certain sub-part fails. More partial syntax trees in the generator's data base would help preventing such fails in the first place. In addition to that, the quality of the generated summary discourse could be improved. At the moment, the generated summaries consist of a series of independently generated statements. Adding support for inter-sentence relations, such as e.g. suggested by Rhetorical Structure Theory, should allow for an even more natural presentation.

Semantic Frames are being developed for more and more languages. While they are language specific, it would be interesting to study how the content representation of a summary could be mapped onto the Frames of another language. This would be a first step toward cross-lingual summaries. It is not clear though if the generation algorithm could be re-used for a such a task without further modification. Naturally, the set of partial syntax trees would have to be created for any new language, but the combination rules for tree might not be reusable in a one-to-one fashion. This would certainly be an interesting extension of our approach.

The summaries generated in the system described in this thesis are general purpose summaries. A useful extension could thus be to allow the user of a summary to specify a particular topic of interest, so that an abstractive summary tailored especially to the user's information need could be generated automatically.

Appendix A

Example of an automatic transcript

The following is an example of an automatically generated transcript of meeting ES2002a from the AMI Corpus. It was generated using the 2007 AMI(DA) system for meeting transcription [Hain et al., 2007a] and cleaned from some of the speech disfluencies with the GroDi tool [Germesin, 2008].

[00:00-00:01]	**B:**	Mm
[00:05-00:08]	**D:**	It's to see how he treats could use the powerpoint presentation
[00:07-00:09]	**B:**	Uh i think it's already on actually
[00:09-00:09]	**D:**	Uh-huh
[00:12-00:13]	**B:**	Yeah
[00:15-00:16]	**B:**	Well i don't know the same
[00:19-00:20]	**A:**	Maybe
[00:21-00:21]	**B:**	Uh
[00:24-00:24]	**A:**	Uh-huh
[00:31-00:31]	**B:**	Uh
[00:33-00:34]	**B:**	Applies and that
[00:38-00:38]	**A:**	Yeah
[00:40-00:41]	**B:**	Okay right
[00:41-00:42]	**B:**	And l.
[00:46-00:46]	**A:**	Do you think
[00:48-00:49]	**A:**	Yes
[00:48-00:49]	**B:**	Yeah
[00:50-00:51]	**B:**	Okay
[00:54-00:54]	**B:**	Right
[00:56-00:59]	**B:**	Um well as the kick-off meeting for our project
[01:02-01:03]	**B:**	And
[01:04-01:07]	**B:**	This is what we're gonna be doing as an extra five minutes
[01:08-01:11]	**B:**	Um so of course we'll just a kind of
[01:11-01:15]	**B:**	Make sure that we all know each other i'm ryan and the project manager
[01:15-01:15]	**D:**	Okay
[01:16-01:17]	**B:**	Two and to introduce yourself to get
[01:17-01:21]	**A:**	Hi i'm david and i'm supposed to be industrial designer

[01:21-01:21]	**B:**	Okay
[01:22-01:24]	**D:**	And i'm andrew and i'm the marketing
[01:26-01:26]	**C:**	Hmm
[01:26-01:27]	**D:**	Experts
[01:27-01:29]	**C:**	And create an easy interface
[01:29-01:31]	**B:**	Great okay
[01:31-01:32]	**B:**	I know
[01:32-01:34]	**B:**	And sir designing a new remote control
[01:35-01:36]	**B:**	And
[01:36-01:38]	**B:**	I'll have to record easier actually
[01:39-01:39]	**B:**	So that's
[01:40-01:42]	**B:**	Do that and encourages know
[01:45-01:46]	**B:**	And you all right on time
[01:49-01:56]	**B:**	Um yes it is a design a new remote control as you can see this be original trendy and user-friendly
[01:57-01:57]	**B:**	Um
[01:58-02:00]	**B:**	So that's kind of or brief
[02:01-02:01]	**B:**	So why
[02:02-02:03]	**B:**	Um
[02:03-02:06]	**B:**	And there are three different stages to the design
[02:06-02:09]	**B:**	And i'm laurie sure what would you guys have already received
[02:09-02:10]	**B:**	Um
[02:11-02:12]	**B:**	In your emails what did you get
[02:13-02:18]	**A:**	Um i just got the project announcement designing a remote control
[02:15-02:16]	**B:**	Uh-huh
[02:19-02:20]	**A:**	And it's
[02:19-02:20]	**D:**	Yeah that's
[02:20-02:22]	**B:**	Is that what everybody calls okay
[02:21-02:22]	**A:**	You think
[02:23-02:23]	**B:**	Um
[02:24-02:27]	**B:**	So we're gonna have individual work and then a meeting about it
[02:28-02:28]	**B:**	And
[02:29-02:31]	**B:**	Repeat that process three times
[02:32-02:33]	**B:**	Um
[02:34-02:37]	**B:**	And at this point we get try out the whiteboard over there
[02:38-02:39]	**B:**	I have
[02:40-02:41]	**B:**	So that
[02:41-02:43]	**B:**	You get to draw your favourite animal and

[02:44-02:46] **B:** Sum up your favourite characteristics of that
[02:46-02:47] **B:** So if you'd to go first
[02:47-02:48] **D:** I want to huh
[02:48-02:49] **B:** Very good i
[02:54-02:55] **D:** Alright
[02:56-02:57] **D:** So
[03:01-03:02] **D:** This one here right
[03:03-03:04] **B:** Uh-huh
[03:04-03:04] **D:** Okay
[03:05-03:06] **D:** Yeah nice
[03:06-03:08] **D:** Alright my favourite animal
[03:10-03:11] **D:** Is why
[03:23-03:24] **D:** A big l.
[03:32-03:33] **D:** Uh
[03:33-03:35] **D:** Okay it's very characteristics that i
[03:36-03:36] **B:** Yeah
[03:36-03:37] **D:** Um right well
[03:38-03:39] **D:** Basically yeah
[03:40-03:42] **D:** High priority for any animal for me is that they
[03:43-03:44] **D:** Be willing to take a lot of
[03:45-03:46] **D:** Physical affection
[03:47-03:49] **D:** From The family
[03:49-03:50] **D:** And
[03:51-03:54] **D:** Yeah it a lot of personality and
[03:56-03:59] **D:** He says and then robust good health this is blue
[03:60-04:00] **D:** The bigger
[04:01-04:02] **D:** I send it to you
[04:03-04:04] **B:** Right
[04:05-04:05] **B:** Okay
[04:05-04:05] **D:** Yeah
[04:08-04:08] **C:** Mm
[04:15-04:17] **C:** Oh right on would be a monkey
[04:26-04:26] **B:** I
[04:28-04:34] **C:** And the small keys already had a one party it's got a real not a one that but with them
[04:35-04:35] **D:** Uh-huh
[04:36-04:36] **A:** Cool
[04:36-04:36] **B:** Right
[04:41-04:42] **A:** It's too much here
[04:46-04:49] **B:** You take as long over this is a light because we have not

[04:49-04:51] **B:** An awful lot to discuss a call it a day
[04:52-04:53] **B:** To fill out your interest anyway
[04:53-04:54] **A:** Okay
[04:56-04:58] **D:** That’s it for the whole lot more about the school
[04:58-05:00] **B:** A kid what i might have to get yep again then
[04:59-04:59] **D:** Okay
[05:01-05:03] **B:** I don’t know minus and i think on the spot yeah
[05:07-05:08] **D:** Impressionist
[05:09-05:10] **A:** Oh cool
[05:12-05:13] **B:** Is that away on
[05:13-05:14] **A:** Yeah yeah
[05:14-05:15] **B:** I
[05:19-05:19] **A:** Um
[05:20-05:20] **A:** Well
[05:22-05:31] **A:** I don’t know it’s just first i might need off the top of my head it’s bigger even this "’cause" i’m allergic to most animals but you channel four issues inaccurate right
[05:28-05:28] **B:** Oh
[05:32-05:35] **A:** Oh oh yeah and i kinda wheels becoming know i
[05:36-05:38] **A:** You’d everything in sight i
[05:38-05:41] **A:** They’re quite harmless and my role than interesting
[05:38-05:39] **D:** Right
[05:42-05:42] **D:** Hmm
[05:42-05:43] **B:** Okay
[05:44-05:45] **B:** I saw the note of interaction that
[05:46-05:47] **B:** Um
[05:49-05:49] **A:** Mm
[05:50-05:51] **D:** Superb stage kind of
[05:52-05:53] **A:** I
[05:54-05:55] **A:** Tales bit bigger
[05:55-05:57] **B:** I actually is a dog as well
[05:58-05:60] **B:** But I just a different kind of dog
[05:60-05:60] **D:** Yeah
[06:00-06:03] **B:** On my favourite animal is my own though look at home
[06:04-06:04] **A:** Mm
[06:04-06:05] **B:** Um
[06:07-06:09] **B:** That doesn’t really look actually
[06:09-06:09] **A:** I
[06:12-06:12] **A:** Oh
[06:13-06:15] **B:** Yeah it’s more a pig actually

[06:15-06:15] **A:** Uh-huh
[06:16-06:16] **B:** Well
[06:17-06:17] **A:** Yeah
[06:18-06:19] **D:** I see a dog in there
[06:20-06:21] **B:** There are the three good idea
[06:23-06:24] **D:** No seriously
[06:24-06:25] **C:** I
[06:28-06:29] **B:** Ah
[06:33-06:34] **D:** What kind is it
[06:35-06:38] **B:** I give the next year old various things
[06:39-06:39] **B:** Um it
[06:40-06:42] **B:** And what they about that
[06:43-06:45] **B:** That's this to suggest that his tail wags
[06:46-06:50] **B:** And it's very friendly interior and orestes to see
[06:50-06:52] **B:** Right kind of affectionate and
[06:52-06:53] **B:** Um
[06:55-06:56] **B:** Uh
[06:57-07:01] **B:** And it's quite we as well you can doesn't take up too much space
[07:01-07:02] **B:** Um
[07:06-07:07] **B:** And
[07:07-07:07] **A:** Oh
[07:08-07:09] **B:** Eh does of anything which is the tail
[07:10-07:10] **A:** Yeah
[07:10-07:12] **B:** As well this is quite music say
[07:11-07:13] **D:** Is you where the this is on to it really see thing
[07:12-07:12] **A:** Mm
[07:14-07:14] **A:** Mm
[07:14-07:16] **B:** I couldn't see it as i she's had a standard
[07:17-07:20] **B:** And it just all this and this get up and start chasing its tail
[07:18-07:18] **A:** Oh
[07:20-07:21] **A:** I
[07:21-07:22] **B:** It's round living room
[07:22-07:23] **A:** And after that
[07:24-07:25] **B:** Yeah so
[07:24-07:27] **D:** Probably when use little he got lots of attention for doing it in
[07:27-07:28] **B:** Yeah maybe
[07:27-07:29] **D:** Has forever been conditioned
[07:29-07:30] **B:** Maybe i
[07:31-07:33] **B:** And where she found this just an here
[07:35-07:36] **B:** Okay

[07:39-07:40] **B:** Yeah
[07:40-07:41] **B:** Meeting next
[07:42-07:43] **B:** Um
[07:45-07:46] **A:** Uh-huh
[07:45-07:49] **B:** Okay and i need to discuss the project finance
[07:48-07:48] **A:** Uh-huh
[07:50-07:51] **B:** Um
[07:52-07:57] **B:** So current a brief we're gonna be selling this remote control for twenty five euro
[07:57-07:58] **B:** Um i'm where
[07:59-07:60] **B:** Aiming to make
[08:00-08:02] **B:** Fifty million euro
[08:03-08:07] **B:** Um so we're gonna be selling this an international scale
[08:07-08:12] **B:** And we do want it to cost any more than twelve fifty euros saying
[08:13-08:15] **B:** Fifty percent of the selling price
[08:16-08:18] **D:** And we discover that again
[08:18-08:18] **B:** Sure
[08:19-08:21] **D:** Um so this
[08:22-08:23] **D:** Ah right yeah
[08:25-08:26] **D:** So cost
[08:27-08:28] **D:** Production cost is
[08:28-08:29] **B:** All see the other
[08:29-08:32] **D:** Twelve fifty the selling price is a wholesaler reach yeah
[08:33-08:34] **D:** Like on the shelf
[08:35-08:35] **B:** I i
[08:36-08:37] **B:** But i mighta
[08:38-08:39] **B:** That's good question
[08:38-08:40] **D:** Our sale our selling or
[08:40-08:43] **B:** I imagine it probably is are sell actually because it's probably up to
[08:42-08:43] **D:** Okay
[08:44-08:45] **B:** The
[08:46-08:48] **B:** The retailer to yeah some of whatever price they want
[08:49-08:50] **B:** Um
[08:52-08:56] **B:** But i don't know i mean do you think the fact that it's going to be sold internationally will have a bearing on
[08:57-08:58] **B:** Hi redesign it at all
[08:59-08:59] **D:** Yes
[08:59-08:60] **B:** Okay well
[09:01-09:02] **B:** Um

[09:05-09:06] **B:** Mm
[09:08-09:14] **D:** All right away i'm wondering if there is and the d. v. d. players if they are zones
[09:14-09:16] **B:** Oh yeah regions and stuff yeah
[09:14-09:15] **D:** Um
[09:16-09:16] **D:** Frequencies or something
[09:17-09:17] **B:** Yeah
[09:18-09:18] **B:** Okay
[09:17-09:20] **D:** Hmm As well as
[09:20-09:23] **D:** Characters different
[09:23-09:25] **D:** Keypad styles and simple
[09:25-09:28] **B:** Yeah well for a remote control think that would be
[09:26-09:27] **D:** Uh mm
[09:28-09:31] **B:** I suppose depends on how complicated a remote control is
[09:29-09:29] **D:** You know
[09:32-09:33] **A:** It does make sense mm
[09:34-09:35] **A:** It is i think it's
[09:36-09:36] **A:** Yeah
[09:37-09:40] **A:** More complicated for your can languages and you need buttons
[09:40-09:41] **B:** Yeah
[09:41-09:41] **D:** Yeah
[09:41-09:42] **B:** Yeah
[09:43-09:44] **B:** Okay
[09:43-09:47] **D:** And then at it and then all of the other thing international is on top of the price
[09:48-09:49] **D:** I'm thinking
[09:51-09:52] **D:** The price might
[09:52-09:57] **D:** Might appeal to a certain market in one region where is in another it would be different
[09:58-10:01] **B:** Or just in terms of the wealth of the country
[09:58-10:00] **D:** Just a characteristic huh
[10:01-10:03] **B:** How much money people have to spend on things that
[10:01-10:01] **D:** Just
[10:03-10:03] **D:** Or just
[10:04-10:06] **D:** Basic product that is positioning
[10:06-10:11] **D:** Twenty five euro remote control might be a big hit in london
[10:11-10:14] **D:** Might not be such a big hit in and
[10:15-10:16] **D:** Grease and
[10:16-10:17] **B:** Yeah it's them yeah
[10:16-10:17] **D:** Something that

[10:17-10:17] **A:** Uh-huh

[10:19-10:23] **B:** Marketing Get more control over it should be writing on this time

[10:23-10:24] **B:** Um

[10:55-10:58] **D:** Right away and making some kind of assumptions about what

[10:59-11:00] **D:** What information we’re getting here

[11:01-11:01] **B:** Mm

[11:01-11:01] **D:** Thinking

[11:02-11:04] **D:** "’kay" trendy probably means something

[11:04-11:05] **D:** Other than just basic

[11:05-11:06] **B:** Yeah

[11:06-11:07] **D:** Something other than just

[11:07-11:08] **D:** Standard

[11:09-11:09] **D:** And

[11:10-11:11] **D:** So

[11:11-11:16] **D:** Wondering right away is selling twenty five years is that is gonna be the premium product

[11:17-11:20] **B:** Yeah yeah how much does remote control cost

[11:17-11:18] **D:** Okay

[11:22-11:24] **B:** Well twenty five euro mean that

[11:25-11:26] **B:** That’s quite

[11:26-11:29] **B:** Eighteen pounds or something isn’t are know as much as well

[11:30-11:31] **B:** Sixteen seventeen eighteen times

[11:30-11:32] **D:** Yeah Yeah it’s yeah

[11:32-11:33] **B:** Um

[11:34-11:36] **B:** As in i’ve never bought a remote controls are you

[11:37-11:39] **B:** Hi how good a remote control that we’ve got you

[11:36-11:37] **D:** No

[11:39-11:40] **B:** Um

[11:41-11:42] **D:** I am

[11:42-11:45] **B:** But yes is it has to look kind of cool and gimmicky

[11:44-11:44] **D:** Uh-huh

[11:47-11:48] **B:** Um

[11:49-11:49] **B:** Right

[11:50-11:50] **B:** Okay

[11:52-11:53] **B:** We just go on ahead here

[11:54-11:54] **B:** Okay

[11:55-11:58] **B:** Uh well that does anybody have anything to add to

[11:58-11:59] **B:** Uhuh

[11:60-12:02] **B:** To the finance issue at all

[12:03-12:07] **D:** Do we have any other background information on how that compares to other
[12:08-12:11] **B:** No actually that would be useful though it may if you knew
[12:08-12:09] **D:** Other mm
[12:11-12:12] **B:** What your money look at you
[12:11-12:11] **D:** Yeah
[12:13-12:13] **B:** Nine
[12:15-12:16] **B:** Mm
[12:34-12:37] **D:** Here interesting thing about discussing the
[12:37-12:39] **D:** Production remote control for me is that
[12:39-12:43] **D:** But as you point out as don't think remote controls being summoned something people
[12:44-12:45] **D:** Consciously
[12:45-12:47] **D:** Assess in the purchasing habits
[12:48-12:48] **B:** Yeah yes
[12:48-12:49] **D:** It's just
[12:50-12:53] **D:** Getting shoelaces issues or something this comes from
[12:53-12:56] **B:** Five minutes and the meeting oh okay river behind
[12:56-12:57] **D:** Tony
[12:58-12:58] **A:** Yeah
[12:57-12:58] **C:** Yeah
[12:58-13:01] **D:** Is it is or how do you i mean one one way look at a be well
[13:02-13:03] **D:** The people producing
[13:03-13:05] **D:** Television set maybe they have to buy
[13:05-13:06] **D:** Remote control
[13:07-13:11] **D:** Another way is maybe people had t. v. set a really set up with their remote control
[13:11-13:12] **D:** And they really want
[13:13-13:13] **C:** And
[13:13-13:14] **D:** A better one or something
[13:14-13:21] **C:** Okay in time but i'm more controls because the fed up of having four five different calls for each thing starts
[13:20-13:21] **D:** Right Right
[13:21-13:24] **C:** So and then was just it's how many choices control
[13:22-13:22] **D:** Okay
[13:24-13:27] **D:** Right so in function wanna priorities might be
[13:26-13:27] **B:** Yeah
[13:27-13:28] **D:** Two
[13:29-13:29] **D:** Combine as many
[13:30-13:30] **D:** Uses

[13:31-13:35] **B:** That's a design that should be a main design a name of a remote control do

[13:35-13:36] **D:** I think yeah

[13:36-13:37] **B:** Your your satellite ten year

[13:37-13:37] **D:** Yeah

[13:38-13:40] **B:** Regular tally in your v. c. r. and everything

[13:40-13:41] **D:** Like to

[13:41-13:45] **D:** Maybe what we could use as a sort example of a successful

[13:45-13:46] **D:** At least technology is

[13:47-13:48] **D:** Palm palm pilots

[13:48-13:54] **D:** They're drawn from being just little scribble boards to the cameras m. p. three players cell phones

[13:53-13:53] **B:** Hmm

[13:55-13:59] **D:** Everything agenda i wonder if we might add something you do that

[13:59-14:03] **D:** To the remote control market such as the lighting in your hair are

[14:00-14:01] **B:** Yeah

[14:03-14:04] **D:** Um

[14:04-14:06] **B:** Or even a in a it's a ball it's

[14:06-14:11] **B:** Um what you wanna watch slightly my pen they're all i want one sentence estimate and that's a good idea

[14:09-14:10] **D:** Yeah yeah

[14:12-14:13] **B:** So a extra functionalities

[14:11-14:11] **D:** And

[14:13-14:22] **D:** Yeah but personally for me at home i've combined be the audio video might television set in my d. v. d. player m. s. c. d. player

[14:23-14:26] **D:** So they will work actually function together but i differ remote controls for each yeah

[14:27-14:27] **B:** Uh-huh is

[14:27-14:30] **D:** So is sort of ironic that then they're in there

[14:32-14:33] **D:** Um

[14:34-14:36] **D:** You know the sound everything is just one system

[14:37-14:37] **B:** Hmm

[14:36-14:38] **D:** But each one's got it on the know

[14:39-14:39] **D:** Right

[14:43-14:44] **B:** Um

[14:45-14:49] **B:** Okay at all gonna have to wrap up pretty quickly a nice couple of minutes

[14:50-14:51] **B:** And i'll just write nothing else

[14:52-14:53] **B:** Okay

[14:54-14:59] **B:** So anything else anybody wants at it but what they don't about remote controls and use what they do

[14:59-15:00] **B:** Would really to be

[15:01-15:03] **B:** Part of this new one at all

[15:03-15:04] **A:** And you using them

[15:05-15:06] **B:** You keep using them at a

[15:06-15:08] **A:** Find some here okay uh-huh

[15:07-15:08] **D:** Mm

[15:09-15:11] **A:** I mean that's really quite small or when you why

[15:09-15:09] **D:** Hmm

[15:11-15:12] **D:** Uh-huh

[15:12-15:13] **A:** Yeah i think

[15:14-15:14] **A:** And you

[15:13-15:14] **B:** Yeah

[15:13-15:14] **D:** Yeah

[15:15-15:19] **B:** You get a response we can if you whistle or make really high pitched noisy beep

[15:15-15:15] **D:** Yeah

[15:16-15:17] **D:** This is really yeah

[15:19-15:20] **D:** Yeah

[15:20-15:23] **B:** But i mean is that something we don't include you think

[15:24-15:25] **D:** Huh

[15:25-15:26] **B:** Hmm

[15:26-15:26] **D:** Sure

[15:27-15:27] **B:** Okay maybe

[15:27-15:30] **D:** And i remember when the first remote control might

[15:30-15:34] **D:** My family had was on a cable actually cable between get the t. v. in big

[15:34-15:35] **D:** Buttons as sort of

[15:36-15:37] **D:** Like on a blind or something

[15:37-15:38] **B:** My goodness

[15:37-15:40] **D:** And anything about what they are now

[15:40-15:42] **D:** Better but actually it's still kind of

[15:43-15:44] **D:** I don't know

[15:44-15:46] **D:** And massive junkie thing on the table

[15:46-15:48] **B:** Stuff is quite primitive

[15:46-15:48] **D:** Maybe we could think about how

[15:49-15:51] **D:** Could be more streamline

[15:51-15:53] **B:** Maybe like a touch screen or something

[15:53-15:56] **D:** Something that yeah or whatever be technologically reasonable

[15:54-15:55] **B:** Okay

[15:56-15:59] **B:** Uh-huh okay well i guess that’s up to our industrial designer

[15:57-16:03] **D:** "’cause" it could be that if it could be the functionally that doesn’t make any better but that just t. p. all of

[16:02-16:03] **B:** Yeah that’s better

[16:03-16:05] **D:** I’ve not having these days is a rip

[16:06-16:08] **D:** Using that he was phones are becoming more and more

[16:08-16:09] **D:** Sheik

[16:09-16:09] **B:** Yeah

[16:10-16:12] **D:** Um nicer material than

[16:12-16:12] **B:** Okay

[16:12-16:13] **D:** Might be

[16:15-16:16] **B:** Okay

[16:16-16:17] **D:** Be worth exploring yeah

[16:17-16:25] **B:** Right well and just to wrap up the next meeting’s gonna be in thirty minutes that’s point about ten to twelve by my watch

[16:25-16:27] **B:** And

[16:27-16:28] **B:** In between nine and

[16:29-16:30] **B:** Um

[16:30-16:33] **B:** As the industrial designer you’re gonna be working on

[16:33-16:36] **B:** You know that’s a working design of that you doing there

[16:37-16:38] **B:** Um

[16:38-16:39] **B:** For our user interface

[16:40-16:44] **B:** Technical functions i guess that’s what we’ve been talking about what i’ll actually day

[16:45-16:46] **B:** And and

[16:47-16:49] **B:** Uh marketing executive

[16:50-16:50] **B:** You’ll be

[16:51-16:54] **B:** Thinking about what it actually what requirements it has to

[16:55-16:56] **B:** Has to fulfil

[16:56-16:59] **B:** And you will get instructions email t. i guess

[16:60-17:00] **D:** Okay

[17:01-17:02] **B:** Um

[17:06-17:07] **B:** Okay

[17:08-17:12] **B:** Yes it’s at the functional design stages next i guess

[17:14-17:15] **B:** And

[17:16-17:18] **B:** And that’s the end of the meeting

[17:19-17:23] **B:** So i got that little message a lock same as my thought i would say

[17:20-17:20] **D:** Um
[17:27-17:30] **D:** Before we wrap up just to make your own same page yeah
[17:30-17:30] **B:** Uh-huh
[17:30-17:31] **D:** Um
[17:31-17:32] **D:** Do we
[17:33-17:37] **D:** We’re given sort of an example of a coffee machine or something right
[17:36-17:37] **B:** Nine yeah
[17:37-17:37] **D:** Well
[17:39-17:39] **D:** Um
[17:41-17:47] **D:** Are we i mean right now nice function a television remote control may have features which could be on the television
[17:48-17:50] **D:** Or are we keeping sort of
[17:50-17:52] **D:** And design commitment to
[17:52-17:53] **D:** Television feature
[17:54-18:02] **B:** Okay well just very quickly "’cause" we’re supposed to finish my and i guess that’s not to us i mean you probably want some kind of unique selling point of that
[17:54-17:55] **D:** I don’t know
[18:02-18:04] **B:** So
[18:02-18:02] **D:** Okay
[18:04-18:07] **A:** I don’t want that it would be production cost
[18:04-18:05] **B:** You know
[18:07-18:08] **D:** Okay yeah
[18:08-18:10] **A:** Because there’s a cat there
[18:08-18:08] **B:** Yeah
[18:10-18:10] **D:** Okay
[18:11-18:13] **A:** Depends on how much you can cram into that price
[18:14-18:14] **B:** Uh-huh
[18:14-18:14] **D:** Okay
[18:16-18:18] **A:** I think that’s the main
[18:16-18:16] **B:** Yeah
[18:18-18:19] **B:** Okay
[18:18-18:19] **D:** Okay
[18:19-18:22] **B:** Right okay well that’s the end of the meeting them
[18:22-18:23] **B:** Um
[18:25-18:27] **B:** Uh Thank you all for coming
[18:25-18:26] **D:** I am
[18:28-18:28] **A:** Yeah
[18:29-18:29] **B:** I

[18:31-18:34]	**B:**	Okay that was what they want us to listen
[18:35-18:36]	**B:**	Um
[18:38-18:39]	**B:**	Right
[18:40-18:40]	**D:**	Great
[18:41-18:42]	**B:**	How is it as well
[18:47-18:47]	**B:**	So
[18:48-18:48]	**B:**	Function and
[18:50-18:50]	**B:**	Okay okay
[18:53-18:54]	**D:**	Mm
[18:54-18:56]	**B:**	I mean if you can just leave it on maybe in them
[18:57-18:57]	**A:**	No
[18:57-18:57]	**B:**	Um
[18:58-18:58]	**A:**	Right well
[18:59-19:01]	**B:**	Oh call this time to sell four
[19:02-19:02]	**C:**	If
[19:04-19:04]	**A:**	Right right okay
[19:04-19:04]	**C:**	Mm
[19:05-19:08]	**A:**	Function mm e. f. eight
[19:09-19:09]	**B:**	Uh-huh
[19:10-19:10]	**C:**	Mm
[19:12-19:14]	**A:**	Yeah it's just something that
[19:12-19:12]	**B:**	Oh yeah that's
[19:14-19:14]	**C:**	Okay
[19:14-19:16]	**B:**	Oh thank goodness okay
[19:17-19:17]	**B:**	Right
[19:30-19:31]	**B:**	Mm
[19:38-19:38]	**B:**	I
[20:19-20:19]	**B:**	Oh
[20:29-20:29]	**B:**	Oh
[20:31-20:32]	**B:**	Mm Oh
[20:34-20:34]	**B:**	Mm
[20:40-20:40]	**B:**	Oh
[20:43-20:43]	**B:**	I
[20:54-20:54]	**B:**	Mm
[21:11-21:11]	**B:**	Mm

Appendix B

Part-of-Speech Tags used in the Penn Treebank Project

[Santorini, 1990]

$	dollar	NNS	noun, common, plural
“	opening quotation mark	PDT	pre-determiner
”	closing quotation mark	POS	genitive marker
(	opening parenthesis	PRP	pronoun, personal
)	closing parenthesis	PRP$	pronoun, possessive
,	comma	RB	adverb
–	dash	RBR	adverb, comparative
.	sentence terminator	RBS	adverb, superlative
:	colon or ellipsis	RP	particle
CC	conjunction, coordinating	SYM	symbol
CD	numeral, cardinal	TO	"to" as preposition or infinitive marker
DT	determiner	UH	interjection
EX	existential there	VB	verb, base form
FW	foreign word	VBD	verb, past tense
IN	preposition or conjunction, subordinating	VBG	verb, present participle or gerund
JJ	adjective or numeral, ordinal	VBN	verb, past participle
JJR	adjective, comparative	VBP	verb, present tense, not 3rd person singular
JJS	adjective, superlative	VBZ	verb, present tense, 3rd person singular
LS	list item marker	WDT	WH-determiner
MD	modal auxiliary	WP	WH-pronoun
NN	noun, common, singular or mass	WP$	WH-pronoun, possessive
NNP	noun, proper, singular	WRB	Wh-adverb
NNPS	noun, proper, plural		

Bibliography

J. Alexandersson. *Hybrid Discourse Modeling and Summarization for a Speech-to-Speech Translation System.* PhD thesis, Universität des Saarlandes, 2003.

J. Alexandersson, N. Reithinger, and E. Maier. Insights into the dialogue processing of verbmobil. In *Proceedings of the fifth conference on Applied natural language processing ANLP '97*, pages 33–40, San Francisco, 1997. Association for Computational Linguistics.

J. Alexandersson, T. Becker, and N. Pfleger. Overlay: The basic operation for discourse processing. In Wahlster [2006], pages 255–267.

I. Androutsopoulos and P. Malakasiotis. A survey of paraphrasing and textual entailment methods. *Journal of Artificial Intelligence Research*, 38:135–187, 2010.

J. Ang, Y. Liu, and E. Shriberg. Automatic Dialog Act Segmentation and Classification in Multiparty Meetings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2005*, volume 1, pages 1061–1064, 2005.

ANSI-96. Guidelines for abstracts. NISO Press, November 27 1996. Bethesda, MD.

C. Aone, M. E. Okurowski, J. Gorlinsky, and B. Larsen. A trainable summarizer with knowledge acquired from robust nlp techniques. In Mani and Maybury [1999], pages 71–80.

Aristotle. *Categories*. Project Gutenberg, 2000. `http://www.gutenberg.org/files/2412/2412-h/2412-h.htm`.

D. Arsić, B. Schuller, and G. Rigoll. Suspicious behavior detection in public transport by fusion of low-level video descriptors. In *Proceedings of the 8th International Conference on Multimedia and Expo (ICME)*, pages 2018–2021, 2007.

J. L. Austin. *How to do things with words.* Harvard University Press, Cambridge, MA, 2nd edition, 1975.

M. Banerjee, M. Capozzoli, L. McSweeney, and D. Sinha. Beyond kappa: A review of interrater agreement measures. *The Canadian Journal of Statistics / La Revue Canadienne de Statistique*, 27(1):3–23, 1999.

C. Barras, E. Geoffrois, Z. Wu, and M. Liberman. Transcriber: Development and use of a tool for assisting speech corpora production. *Speech Communication*, 33(1–2):5–22, January 2001.

R. Barzilay and M. Elhadad. Using lexical chains for text summarization. In *Proceedings of the ACL Workshop on Intelligent Scalable Text Summarization*, pages 10–17, 1997.

J. Besser. A corpus-based approach to the classification and correction of disfluencies in spontaneous speech. Bachelor's thesis, Universität des Saarlandes, November 2006.

BNC07. The British National Corpus, version 3 (BNC XML Edition). Oxford University Computing Services on behalf of the BNC Consortium, 2007. `http://www.natcorp.ox.ac.uk/`.

H. C. Boas, E. Ponvert, M. Guajardo, and S. Rao. The current status of German FrameNet. In *Multi-lingual semantic annotation: Theory and applications*, Saarbrücken, Germany, June 2006.

H. Borko and C. L. Bernier. *Abstracting concepts and methods.* Academic Press, London, 1975.

H. Bunt. Context and dialogue control. *Think*, 3:19–31, 1994.

A. Burchardt, K. Erk, A. Frank, A. Kowalski, S. Padó, and M. Pinkal. The SALSA Corpus: a German Corpus Resource for Lexical Semantics. In *Proceedings of LREC 2006*, pages 969–974, Genoa, Italy, 2006.

S. Castronovo. Robuste Analyse des Diskussionsstandes von Gruppenbesprechungen mit Hilfe eines wissensbasierten Diskursgedächtnisses. Master's thesis, Saarland University, 2009.

S. Castronovo, J. Frey, and P. Poller. A generic layout-tool for summaries of meetings in a constraint-based approach. In A. Popescu-Belis and R. Stiefelhagen, editors, *Machine Learning for Multimodal Interaction (MLMI-08)*, volume 5237 of *Lecture Notes in Computer Science, LNCS*, pages 248–259. Springer, Heidelberg, 2008. ISBN 978-3-540-85852-2.

E. Charniak and M. Johnson. Coarse-to-fine *n*-best parsing and MaxEnt discriminative reranking. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 173–180, 2005.

J. Cohen. A coefficient of agreement for nominal scales. *Educational and psychological measurement*, 20:37–46, 1960.

E. T. Cremmins. *The art of abstracting.* Information Resources Press, Arlington, VA, 2nd edition, 1996.

D. A. Cruse. *Lexical semantics.* Cambridge University Press, Cambridge, UK, 1986.

D. Das and N. A. Smith. Semi-supervised frame-semantic parsing for unknown predicates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 1435–1444, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P11-1144`.

D. Das, N. Schneider, D. Chen, and N. A. Smith. Probabilistic frame-semantic parsing. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 948–956, Los Angeles, California, June 2010. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N10-1138`.

I. M. R. De Bleecker. Towards an optimal lexicalization in a natural-sounding portable natural language generator for dialog systems. In *Proceedings of the ACL Student Research Workshop*, pages 61–66, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics.

S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41:391–407, 1990.

G. DeJong. Prediction and substantiation: A new approach to natural language processing. *Cognitive Science*, 3(4):251–271, 1979.

G. DeJong. An Overview of the FRUMP System. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for Natural Language Processing*, pages 149–176. Lawrence Erlbaum, Hillsdale, NJ, 1982.

A. Dielmann and S. Renals. DBN based joint dialogue act recognition of multiparty meetings. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007*, pages IV–133–IV–136, Honolulu, HI, April 2007.

H. P. Edmundson. New methods in automatic extracting. *Journal of the ACM*, 16(2):264–285, 1969. ISSN 0004-5411. doi: http://doi.acm.org/10.1145/321510.321519.

B. Endres-Niggemeyer. *Summarizing Information.* Springer, Berlin Heidelberg, 1998. ISBN 3-540-63735-4.

U. Endriss. Semantik zeitlicher Ausdrücke in Terminvereinbarungsdialogen. Technical report, Technische Universität Berlin, 1998. Verbmobil-Report No. 227.

R. Engel. Spin: A semantic parser for spoken dialog systems. In *Proceedings of the Fifth Slovenian And First International Language Technology Conference (IS-LTC 2006)*, Ljubljana, Slovenia, 2006.

K. Erk and S. Padó. Shalmaneser–a toolchain for shallow semantic parsing. In *Proceedings of LREC 2006*, pages 527–532, Genoa, Italy, May 2006.

S. Evert, J. Carletta, T. J. O'Donnell, J. Kilgour, A. Vögele, and H. Voormann. The NITE Object Model. Technical report, University of Edinburgh, March 24 2003. `www.ltg.ed.ac.uk/NITE/documents/NiteObjectModel.v2.1.pdf`.

C. Fellbaum, editor. *WordNet–An Electronic Lexical Database.* MIT Press, Cambridge, MA, May 15 1998.

C. J. Fillmore. The case for case. In E. Bach and R. T. Harms, editors, *Universals in linguistic theory*. Holt, Rinehart and Winston, New York, 1968.

C. J. Fillmore. Frame semantics and the nature of language. *Annals of the New York Academy of Sciences: Conference on the Origin and Development of Language and Speech*, 280:20–32, 1976.

C. J. Fillmore and C. F. Baker. A frames approach to semantic analysis. In B. Heine and H. Narrog, editors, *Oxford Handbook of Linguistic Analysis*, pages 313–340. OUP, Oxford, UK, 2010.

C. J. Fillmore and M. R. Petruck. Framenet glossary. *International Journal of Lexicography*, 16(359–361), 2003.

W. Finkler. *Automatische Selbstkorrektur bei der inkrementellen Generierung gesprochener Sprache unter Realzeitbedingungen: Ein empirisch-simulativer Ansatz unter Verwendung eines Begründungsverwaltungssystems.* Number 165 in DISKI. Infix, 1997.

S. Germesin. Disfluency classification and correction with a hybrid machine learning and rule-based approach. Master's thesis, Universität des Saarlandes, 2008.

S. Germesin, T. Becker, and P. Poller. Determining latency for online dialog act classification. In *Proceedings of MLMI'08*, Utrecht, The Netherlands, 2008.

D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

J. Godfrey, E. Holliman, and J. McDaniel. Switchboard: Telephone speech corpus for research and development. In *Proceedings of ICASSP-92*, pages 517–520, 1992.

E. Goldberg, N. Driedger, and R. I. Kittredge. Using natural-language processing to produce weather forecasts. *IEEE Expert / IEEE Intelligent Systems - EXPERT*, 9(2):45–53, 1994.

P. Grenon. BFO in a Nutshell: A Bi-categorial Axiomatization of BFO and Comparison with DOLCE. `http://www.ifomis.org/bfo`, 2003.

T. R. Gruber. A translation approach to portable ontology specifications. *Knowledge Acquisition*, 5:199–220, 1993.

T. Hain, L. Burget, J. Dines, G. Garau, M. Karafiat, D. V. Leeuwen, M. Lincoln, and V. Wan. The 2007 AMI(DA) system for meeting transcription. In *Proceedings of Classification of Events, Activities and Relationships–CLEAR*, pages 414–428, 2007a.

T. Hain, V. Wan, L. Burget, M. Karafiat, J. Dines, J. Vepa, G. Garau, and M. Lincoln. The ami system for the transcription of speech in meetings. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2007.*, volume 4, pages IV–357–IV–360, April 15–20 2007b.

M. A. K. Halliday and R. Hasan. *Cohesion in English.* Longman, London, 1976.

G. Herzog and P. Wazinski. Visual translator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review*, 8(2/3):175–187, 1994.

E. Hovy and C.-Y. Lin. Automated Text Summarization in SUMMARIST. In Mani and Maybury [1999], pages 81–94.

E. Hovy, C.-Y. Lin, and L. Zhou. Evaluating duc 2005 using basic elements. In *Proceedings of DUC-2005*, 2005.

E. Hovy, C.-Y. Lin, L. Zhou, and J. Fukumoto. Automated summarization evaluation with basic elements. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC) 2006*, pages 899–902, 2006.

P.-Y. S. Hsueh. *Decision Detection and Tracking: Multimodal Information Fusion from Multiparty Discourse*. PhD thesis, School of Informatics, University of Edinburgh, 2008.

A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters. The ICSI Meeting Corpus. In *ICASSP-2003*, volume 1, pages I–364 – I–367, Hong Kong, April 2003.

A. Janin, J. Ang, S. Bhagat, R. Dhillon, J. Edwards, J. Macias-Guarasa, N. Morgan, B. Peskin, E. Shriberg, A. Stolcke, C. Wooters, and B. Wrede. The ICSI Meeting Project: Resources and research. In *Proceedings of the NIST ICASSP 2004 Meeting Recognition Workshop*, Montreal, May 2004.

R. Johansson and P. Nugues. LTH: Semantic Structure Extraction using Nonprojective Dependency Trees. In *Proceedings of the Fourth International Workshop on Semantic Evaluations (SemEval-2007)*, 2007.

D. Jones, F. Wolf, E. Gibson, E. Williams, E. Fedorenko, D. Reynolds, and M. Zissman. Measuring the readability of automatic speech-to-text transcripts. In *Proceedings of the 8th European Conference on Speech Communication and Technology (EUROSPEECH-2003)*, pages 1585–1588, Geneva, Switzerland, September 1–4 2003.

K. S. Jones. A statistical interpretation of term specificity and its application in retrieval. *Journal of Documentation*, 28(1):11–21, 1972.

D. Jurafsky, R. Bates, N. Coccaro, R. Martin, M. Meteer, K. R. E. Shriberg, A. Stolcke, P. Taylor, and C. V. Ess-Dykema. Switchboard discourse language modeling project (final report). In *Proceedings of Johns Hopkins LVCSR Workshop-97*, 1997.

D. Klein and C. D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003.

T. Kleinbauer and S. Germesin. ARKTiS - A Fast Tag Recommender System Based On Heuristics. In *Proceedings of the European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD)*, Bled, Slovenia, September, 8–11 2009.

T. Kleinbauer and G. Murray. *From signal processing to multimodal understanding: automatic analysis of human interaction in meetings.*, chapter Summarization. Cambridge University Press, Cambridge, UK, 2012.

T. Kleinbauer, S. Becker, and T. Becker. Combining multiple information layers for the automatic generation of indicative meeting abstracts. In *Proceedings of 11th European Workshop on Natural Language Generation (ENLG07)*, Dagstuhl, Germany, June 17th-20th 2007a.

T. Kleinbauer, S. Becker, and T. Becker. Indicative abstractive summaries of meetings. In *Proceedings of 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, Brno, Czech Republic, June 28th-30th 2007b.

J. Kolář. *Automatic segmentation of speech into sentence-like units.* PhD thesis, Faculty of Applied Sciences, University of West Bohemia, Pilsen, Czech Republic, 2008.

J. Kupiec, J. Pedersen, and F. Chen. A trainable document summarizer. In *Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 68–73, Seattle, WA, 1995.

F. W. Lancaster. *Indexing and Abstracting in Theory and Practice.* University of Illinois, Graduate School of Library and Information Science, 2nd edition, 1998.

C. Lauer, J. Frey, B. Lang, J. Alexandersson, T. Becker, T. Kleinbauer, and H. Lochert. Amigram–a general-purpose tool for multimodal corpus annotation. In *Proceedings of 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, Edinburgh, UK, July 11–13 2005.

W. G. Lehnert. Plot units: A narrative summarization strategy. In W. G. Lehnert and M. H. Ringle, editors, *Strategies for Natural Language Processing.* Lawrence Erlbaum Associates, New Jersey, 1981.

S. Lesch. Classification of multidimensional dialogue acts using maximum entropy. Master's thesis, Saarland University, 2005.

S. Lesch, T. Kleinbauer, and J. Alexandersson. A new metric for the evaluation of dialog act classification. In *Proceedings of 9th Workshop on the semantics and pragmatics of dialogue, Dialor*, Nancy, France, June 9–11 2005.

R. Levy and G. Andrew. Tregex and tsurgeon: Tools for querying and manipulating tree data structures. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 2006)*, 2006.

C.-Y. Lin. ROUGE: A package for automatic evaluation of summaries. In M.-F. Moens and S. Szpakowicz, editors, *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain, 2004.

C.-Y. Lin. *Robust Automated Topic Identification.* PhD thesis, University of Southern California, 1997.

K. Litkowski and O. Hargraves. The preposition project. In *Proceedings of the 2nd ACL-SIGSEM Workshop on The Linguistic Dimensions of Prepositions and their Use in Computational Linguistics Formalisms and Applications*, Colchester, England, April 19–21 2005.

H. Lochert, J. Alexandersson, T. Becker, and T. Kleinbauer. Ontomatters–ontology-based representation of multi-modal multi-party interaction. In *Proceedings of the 2nd Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithm (MLMI)*, Edinburgh, UK, 11–13 July 2005.

J. Lorhard. *Ogdoas Scholastica, continens Diagraphen Typicam artium: Grammatices (Latinae, Graecae), Logices, Rhetorices, Astronomices, Ethices, Physices, Metaphysices, seu Ontologiae–ex praestantium hujus temporis virorum lucubrationibus, pro doctrinae & virtutum studios a juventute.* Apud Georgium Straub, Sangalli (St. Gallen, Switzerland), 1606.

H. Luhn. The automatic creation of literature abstracts. *IBM Journal of Research and Development*, 2:159–165, 1958.

I. Mani. *Automatic Summarization.* John Benjamins Publishing, Amsterdam/Philadelphia, 2001.

I. Mani and M. T. Maybury, editors. *Advances in Automatic Text Summarization.* MIT Press, Cambridge, MA, 1999.

W. Mann and S. Thompson. Rhetorical structure theory: Toward a functional theory of text organization. *Text*, 8(3):243–281, 1988.

D. Marcu. *The Rhetorical Parsing, Summarization, and Generation of Natural Language Texts.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, CA, December 1997.

M. P. Marcus, B. Santorini, and M. A. Marcinkiewicz. Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, 19:313–330, June 1994.

L. Màrquez, K. C. Litkowski, S. Stevenson, and X. Carreras, editors. *Computational Linguistics Special Issue on Semantic Role Labeling*, volume 34. MIT Press, 2008.

C. Masolo, S. Borgo, A. Gangemi, N. Guarino, and A. Oltramari. Wonderweb deliverable d18–ontology library (final). `http://www.loa-cnr.it/Papers/D18.pdf`, December 2003.

T. Matsuzaki and J. Tsujii. Comparative parser performance analysis across grammar frameworks through automatic tree conversion using synchronous grammars. In *Proceedings of COLING 2008*, 2008.

M. T. Maybury. Generating summaries from event data. *Information Processing and Management*, 31(5):735–751, 1995.

K. R. McKeown, J. Robin, and K. Kukich. Generating concise natural language summaries. *Information Processing & Management*, 31(5):702–733, 1995.

G. A. Miller. The magical number seven, plus or minus two: Some limits on our capacity for processing information. *The Psychological Review*, 63:81–97, 1956.

M. Minsky. A framework for representing knowledge. In P. Winston, editor, *The Psychology of Computer Vision*, pages 211–277. McGraw-Hill, New York, 1975.

R. Mitkov. Outstanding issues in anaphora resolution. In *Computational Linguistics and Intelligent Text Processing*, volume 2004/2001 of *Lecture Notes in Computer Science*, pages 110–125. Springer, International, 2001.

N. Morgan, D. Baron, J. Edwards, D. Ellis, D. Gelbart, A. Janin, T. Pfau, E. Shriberg, and A. Stolcke. The Meeting Project at ICSI. In *Proceedings of the First International Human Language Technologies Conference (HLT01)*, San Diego, March 2001.

C. Müller. Automatic recognition of speakers age and gender on the basis of empirical studies. In *Proceedings of the 9th International Conference on Spoken Language Processing (Interspeech 2006 – ICSLP)*, 2006.

G. Murray. *Using Speech-Specific Characteristics for Automatic Speech Summarization*. PhD thesis, University of Edinburgh, 2008.

G. Murray and S. Renals. Term-weighting for summarization of multi-party spoken dialogues. In *Proceedings of 4th Joint Workshop on Multimodal Interaction and Related Machine Learning Algorithms (MLMI)*, pages 155–166, Brno, Czech Republic, 2007.

G. Murray, S. Renals, J. Carletta, and J. Moore. Evaluating automatic summaries of meeting recordings. In *Proceedings of ACL 2005 MTSE workshop*, Ann Arbor, MI, USA, June 2005.

G. Murray, T. Kleinbauer, P. Poller, T. Becker, S. Renals, and J. Kilgour. Extrinsic summarization evaluation: A decision audit task. *ACM Transactions on Speech and Language Processing (TLSP)*, 6(2), October 2009.

G. Murray, G. Carenini, and R. Ng. Interpretation and transformation for abstracting conversations. In *Proceedings of NAACL HLT 2010*, Los Angeles, USA, June 2010.

S. H. Myaeng and D.-H. Jang. Development and evaluation of a statistically-based document summarization system. In Mani and Maybury [1999], pages 61–70.

A. Nenkova. Entity-driven rewrite for multi-document summarization. In *The Third International Joint Conference on Natural Language Processing (IJCNLP08)*, Hyderabad, India, 2008.

A. Nenkova and R. Passonneau. Evaluating content selection in summarization: The pyramid method. In *Proceedings of HLT-NAACL*, pages 145–152, 2004.

H. op den Akker and C. Schulz. Exploring features and classifiers for dialogue act segmentation. In *Machine Learning for Multimodal Interaction*, volume 5237/2008 of *Lecture Notes in Computer Science*, pages 196–207, 2008.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA, 2002. Association for Computational Linguistics.

M. R. L. Petruck. Frame semantics. In J. Verschueren, J.-O. Östman, J. Blommaert, and C. Bulcaen, editors, *Handbook of Pragmatics: 1996 Installment*. John Benjamins, Amsterdam/Philadelphia, 1997.

A. Popescu-Belis. Abstracting a dialogue act tagset for meeting processing. In *Proceedings of 4th International Conference on Language Resources and Evaluation, LREC 2004*, volume IV, pages 1415–1418, Lisbon, Portugal, 2004.

S. S. Pradhan, W. Ward, K. Hacioglu, J. H. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology Conference/North American chapter of the Association for Computational Linguistics, HLT/NAACL-2004*, Boston, MA, May 2–7 2004.

M. Purver, J. Dowding, J. Niekrasz, P. Ehlen, S. Noorbaloochi, and S. Peters. Detecting and summarizing action items in multi-party dialogue. In *Proceedings of the 8th SIGdial Workshop on Discourse and Dialogue*, pages 18–25, Antwerp, Belgium, September 2007.

D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation and user studies. In *Proceedings of the ANLP/NAACL 2000 Workshop*, pages 21–29, Seattle, WA, 2000.

G. J. Rath, A. Resnick, and T. R. Savage. The formation of abstracts by the selection of sentences–part 1: Sentence selection by man and machines. *American Documentation*, 12(2):139–141, 1961.

E. Reiter and R. Dale. *Building natural language generation systems.* Studies in natural language processing. Cambridge University Press, Cambridge, UK, 2000.

N. Reithinger and M. Klesen. Dialogue act classification using language models. In *In Proceedings of EuroSpeech-97*, pages 2235–2238, 1997.

N. Reithinger, M. Kipp, R. Engel, and J. Alexandersson. Summarizing multilingual spoken negotiation dialogues. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics, ACL-2000*, pages 310–317, Hong Kong, October 2000.

J. Ruppenhofer, M. Ellsworth, M. R. L. Petruck, C. R. Johnson, and J. Scheffczyk. FrameNet II: Extended theory and practice. `http://framenet.icsi.berkeley.edu/index.php?option=com_wrapper&Itemid=126`, August 25 2006.

G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.

B. Santorini. Part-of-speech tagging guidelines for the Penn Treebank Project. Technical Report MS-CIS-90-47, Department of Computer and Information Science, University of Pennsylvania, 1990.

R. C. Schank. Conceptual dependency: A theory of natural language understanding. *Cognitive Psychology*, 3(4):552–631, October 1972.

R. C. Schank. *Conceptual information processing.* North Holland, Amsterdam, 1975.

R. C. Schank and R. P. Abelson. *Scripts plans goals and understanding.* Erlbaum, Hillsdale, NJ, 1977.

G. Shafer. Perspectives on the theory and practice of belief functions. *International Journal of Approximate Reasoning*, pages 1–40, 1990.

E. Shriberg, A. Stolcke, D. Jurafsky, N. Coccaro, M. Meteer, R. Bates, P. Taylor, K. Ries, R. Martin, and C. van Ess-Dykema. Can prosody aid the automatic classification of dialog acts in conversational speech? *Language and Speech*, 41(3-4):439–487, 1998.

E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey. The ICSI Meeting Recorder Dialog Act (MRDA) Corpus. In *Proceedings of the 5th SIGdial Workshop on Discourse and Dialogue*, pages 97–100, 2004.

E. E. Shriberg. *Preliminaries to a Theory of Speech Disfluencies.* PhD thesis, University of California at Berkeley, 1994.

K. Sparck Jones. Automatic summarization: factors and directions. In Mani and Maybury [1999], pages 1–12.

A. Stent. Rhetorical structure in dialog. In *Proceedings of the First International Conference on Natural Language Generation (INLG 2000)*, pages 247–252, Mitzpe Ramon, Israel, June 2000. Association for Computational Linguistics.

A. Stolcke, K. Ries, N. Coccaro, E. Shriberg, R. Bates, D. Jurafsky, P. Taylor, R. Martin, C. V. Ess-Dykema, and M. Meteer. Dialogue act modeling for automatic tagging and recognition of conversational speech. *Computational Linguistics*, 26(3):339–371, 2000.

T. Strzalkowski, J. Wang, and B. Wise. A robust practical text summarization. In *AAAI Spring Symposium on Intelligent Text Summarization*, pages 26–33, Stanford, March 1998. AAAI Press.

C. Subirats and M. R. Petruck. Surprise: Spanish FrameNet! In *International Congress of Linguists. Workshop on Frame Semantics*, Prague, Czech Republic, July 2003. Matfyzpress.

S. Teufel and M. Moens. Argumentative classification of extracted sentences as a first step towards flexible abstracting. In Mani and Maybury [1999], pages 155–171.

S. Tratz and E. Hovy. Summarization evaluation using transformed basic elements. In *Proceedings of the 1st Text Analysis Conference (TAC)*, Gaithersburg, MD, 2008. NIST.

S. Tucker and S. Whittaker. Temporal compression of speech: An evaluation. *IEEE Transactions on Audio, Speech and Language Processing*, 16(4):790–796, 2008.

R. Valenza, T. Robinson, M. Hickey, and R. Tucker. Summarization of spoken audio through information extraction. In *Proc. ESCA Workshop on Accessing Information in Spoken Audio*, pages 111–116, 1999.

T. A. van Dijk. *MACROSTRUCTURES–An Interdisciplinary Study of Global Structures in Discourse, Interaction, and Cognition.* Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1980.

T. A. van Dijk and W. Kintsch. *Strategies of Discourse Comprehension.* Academic Press, New York, August 1983.

H. van Halteren and S. Teufel. Examining the consensus between human summaries: initial experiments with factoid analysis. In *Proceedings of the HLT-NAACL 2003 Text Summarization Workshop*, pages 57–64, Stroudsburg, PA, 2003. Association of Computational Linguistics.

C. J. Van Rijsbergen. *Information Retrieval.* Butterworths, London, 2nd edition, 1979. http://citeseer.csail.mit.edu/vanrijsbergen79information.html.

W. Wahlster, editor. *Verbmobil: Foundations of Speech-to-Speech Translation.* Springer, Berlin, 2000.

W. Wahlster, editor. *SmartKom: Foundations of Multimodal Dialogue Systems.* Springer Berlin Heidelberg, 2006.

A. Waibel, M. Bett, M. Finke, and R. Stiefelhagen. Meeting browser: Tracking and summarizing meetings. In *In Proceedings of the DARPA Broadcast News Workshop*, pages 281–286. Morgan Kaufmann, 1998.

L. Wang and C. Cardie. Summarizing decisions in spoken meetings. In *Proceedings of the Workshop on Automatic Summarization for Different Genres, Media, and Languages*, pages 16–24, Portland, Oregon, June 2011. Association for Computational Linguistics.

B. Wrede and E. Shriberg. Spotting "hot spots" in meetings: Human judgments and prosodic cues. In *Proceedings of the 8th European Conference on Speech Communication and Technology (Eurospeech)*, pages 2805–2808, 2003.

M.-H. Yang, D. Kriegman, and N. Ahuja. Detecting faces in images: A survey. *IEEE Transascions on Pattern Analysis and Machine Intelligence*, 24(1):34–58, 2002.

K. Zechner. *Automatic Summarization of Spoken Dialogue in Unrestricted Domains.* PhD thesis, Carnegie Mellon University, 2001.

K. Zechner and A. Waibel. Minimizing word error rate in textual summaries of spoken language. In *Proceedings of the First Meeting of the North American Chapter of the Association for Computational Linguistics, NAACL-2000*, pages 186–193, 2000.

M. Zobl, F. Wallhoff, and G. Rigoll. Action recognition in meeting scenarios using global motion features. In J. Ferryman, editor, *Proceedings of the 4th IEEE International Workshop on Performance Evaluation of Tracking and Surveillance (PETS-ICVS)*, pages 32–36, 2003.

I. Zukerman, E. Makalic, M. Niemann, and S. George. A probabilistic approach to the interpretation of spoken utterances. In *PRICAI 2008: Trends in Artificial Intelligence*, volume 5351/2008 of *Lecture Notes in Computer Science*, pages 581–592, Berlin Heidelberg, 2008. Springer.