

# POPULATING KNOWLEDGE BASES WITH TEMPORAL INFORMATION

---

Thesis for obtaining the title of  
Doctor of Engineering  
of the Faculty of Mathematics and Computer Science  
of Saarland University

by

ERDAL KUZHEY, (M.Sc.)

Saarbrücken  
October, 2016

Day of Colloquium

Dean of the Faculty

Chair of the Committee

Reporters

First reviewer

Second reviewer

Third reviewer

Academic Assistant

28 / 02 / 2017

Univ.-Prof. Dr. Frank-Olaf Schreyer

Prof. Dr. Diettrich Klakow

Prof. Dr. Gerhard Weikum

Prof. Dr. Maarten de Rijke

Prof. Dr. Fabian M. Suchanek

Dr. Luciano Del Corro



*To all members of my family...*



*“I am neither Christian, nor Jew, nor Hindu, nor Moslem.  
I am not of the East, nor of the West, nor of the land, nor of the sea;  
I am not of Nature’s mint, nor of the circling heavens.  
I am not of earth, nor of water, nor of air, nor of fire;  
I am not of the empyrean, nor of the dust, nor of existence, nor of entity.  
I am not of this world, nor of the next, nor of Paradise, nor of Hell;  
I am not of Adam and Eve, nor of any origin story.  
My place is the Placeless, my trace is the Traceless;  
Neither body nor soul, I am of the Divine Whole.  
I belong to the beloved.  
...  
I am nothing but a breath.”*

Rumi

*“Ben ezelden beridir hür yaşadım, hür yaşarım,  
Hangi çılgın bana zincir vuracakmış? Şaşarım.  
Kükremiş sel gibiyim, bendimi çiğner, aşarım,  
Yırtarım dağları, enginlere sığmam, taşarım.”*

M. Âkif Ersoy



## *Acknowledgements*

I would not have been able to finish this dissertation without the support of many people.

First and foremost, I am deeply grateful to Gerhard Weikum. I feel fortunate to be his student. His great support, guidance, encouragement, and patience during my doctoral studies were invaluable and motivated me for pursuing this dissertation. He is not only a great scientist, but also a great teacher and leader. He always made me feel that I can count on him.

I also thank Jilles Vreeken, Vinay Setty, Jannik Strötgen, and Fabian Suchanek for the valuable work we published together.

I would like to express my sincere gratitude to my colleagues in the Databases and Information Systems group at MPII. I learned a lot from them during “Wednesday group meetings” and also the stimulating discussions in the department kitchen. Particularly, I thank Rawia Awadallah, Steffen Metzger, Asia Biega, Niket Tandon, Christina Teflioudi, Saskia Metzler, and Luciano Del Corro.

I thank Ricarda Dubral and Holger Dell for helping me with the German translation of the abstract of this dissertation, and Andrew Yates for reviewing the Introduction chapter.

I wish to thank Cinzia di Ubaldo for her support in the early stages of my doctoral studies.

I thank the close circle of friends who made me enjoy the life in Saarbrücken: Yağız Kargın, Ramazan Ayaslı, Doğan Karaoğlan, Christina Teflioudi, Vladimir Bessonov, Dominik Cermann, Güneş Oba, Zeynep Köylüoğlu, İpek Atıla.

I specially thank to my life coach and best friend Yağız Kargın for his constant support during my difficult days. His spiritual advices enabled me to have a calm mind and to enjoy the very current moment, now.

I am grateful to Ricarda Dubral. Her love has been nurturing me since I met her.

My parents, Makbule and Hasan, have been a great blessing for me and my siblings. Their confidence in me let me grow up spiritually and mentally. Their simple parenting based on love and care is the fundamental of what I all have. I love them.





# Abstract

Recent progress in information extraction has enabled the automatic construction of large knowledge bases. Knowledge bases contain millions of entities (e.g. persons, organizations, events, etc.), their semantic classes, and facts about them. Knowledge bases have become a great asset for semantic search, entity linking, deep analytics, and question answering. However, a common limitation of current knowledge bases is the poor coverage of temporal knowledge. First of all, so far, knowledge bases have focused on popular events and ignored long tail events such as political scandals, local festivals, or protests. Secondly, they do not cover the textual phrases denoting events and temporal facts at all.

The goal of this dissertation, thus, is to automatically populate knowledge bases with this kind of temporal knowledge. The dissertation makes the following contributions to address the afore mentioned limitations. The first contribution is a method for extracting events from news articles. The method reconciles the extracted events into canonicalized representations and organizes them into fine-grained semantic classes. The second contribution is a method for mining the textual phrases denoting the events and facts. The method infers the temporal scopes of these phrases and maps them to a knowledge base.

Our experimental evaluations demonstrate that our methods yield high quality output compared to state-of-the-art approaches, and can indeed populate knowledge bases with temporal knowledge.



# Kurzfassung

Der Fortschritt in der Informationsextraktion ermöglicht heute das automatische Erstellen von Wissensbasen. Derartige Wissensbasen enthalten Entitäten wie Personen, Organisationen oder Events sowie Informationen über diese und deren semantische Klasse. Automatisch generierte Wissensbasen bilden eine wesentliche Grundlage für das semantische Suchen, das Verknüpfen von Entitäten, die Textanalyse und für natürlichsprachliche Frage-Antwortsysteme. Eine Schwäche aktueller Wissensbasen ist jedoch die unzureichende Erfassung von temporalen Informationen. Wissensbasen fokussieren in erster Linie auf populäre Events und ignorieren weniger bekannte Events wie z.B. politische Skandale, lokale Veranstaltungen oder Demonstrationen. Zudem werden Textphrasen zur Bezeichnung von Events und temporalen Fakten nicht erfasst.

Ziel der vorliegenden Arbeit ist es, Methoden zu entwickeln, die temporales Wissen automatisch in Wissensbasen integrieren. Dazu leistet die Dissertation folgende Beiträge:

1. Die Entwicklung einer Methode zur Extrahierung von Events aus Nachrichtenartikeln sowie deren Darstellung in einer kanonischen Form und ihrer Einordnung in detaillierte semantische Klassen.
2. Die Entwicklung einer Methode zur Gewinnung von Textphrasen, die Events und Fakten in Wissensbasen bezeichnen sowie einer Methode zur Ableitung ihres zeitlichen Verlaufs und ihrer Dauer.

Unsere Experimente belegen, dass die von uns entwickelten Methoden zu qualitativ deutlich besseren Ausgabewerten führen als bisherige Verfahren und Wissensbasen tatsächlich um temporales Wissen erweitern können.



# CONTENTS

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Goals and Challenges . . . . .	2
1.3	Contributions . . . . .	4
1.4	Dissertation Outline . . . . .	6
<b>2</b>	<b>Background &amp; Related Work</b>	<b>7</b>
2.1	Knowledge Base Preliminaries . . . . .	7
2.2	Temporal Knowledge . . . . .	11
2.3	Information Extraction . . . . .	13
2.4	Related Tasks . . . . .	17
2.5	Summary . . . . .	19
<b>3</b>	<b>Populating Knowledge Bases with Events</b>	<b>21</b>
3.1	Motivation . . . . .	21
3.2	Contribution . . . . .	24
3.3	Related Work . . . . .	26
3.4	System Overview . . . . .	27
3.5	Features and Distance Measures . . . . .	29
3.6	Computing Semantic Types for News . . . . .	31
3.7	Multi-view Attributed Graph (MVAG) . . . . .	36
3.8	From News to Events . . . . .	37
3.9	Evaluation . . . . .	52
3.10	Applications . . . . .	66
3.11	Summary . . . . .	69
<b>4</b>	<b>Populating Knowledge Bases with Temponyms</b>	<b>71</b>
4.1	Motivation . . . . .	71
4.2	Approach and Contribution . . . . .	76
4.3	Prior Work and Background . . . . .	78
4.4	System Overview . . . . .	81
4.5	Temponym Detection . . . . .	86

---

4.6	Candidate Mappings Generation . . . . .	90
4.7	Temponym Disambiguation . . . . .	91
4.8	Populating the KB with temponyms. . . . .	99
4.9	Evaluation . . . . .	99
4.10	Applications . . . . .	110
4.11	Summary . . . . .	113
<b>5</b>	<b>Conclusion</b>	<b>115</b>
5.1	Summary . . . . .	115
5.2	Outlook . . . . .	116
	<b>List of Figures</b>	<b>117</b>
	<b>List of Tables</b>	<b>119</b>
	<b>Bibliography</b>	<b>121</b>
	<b>Index</b>	<b>141</b>

# CHAPTER 1

## INTRODUCTION

### 1.1 MOTIVATION

In the context of computers, knowledge is the compilation of facts, descriptions of and information about things [1]. Therefore, it is crucial to gather this knowledge and store it in a machine-readable format. Such a store of knowledge is called a *knowledge base (KB)*. KB's contain entities (e.g. people, organizations, countries, events, etc.), their alias names, their semantic classes, the relationships among them, and factual assertions about them. Prominent examples of KB's are YAGO, Freebase, DBpedia, and Wikidata. KB's are a key resource enabling computers to perform cognitive applications like semantic search, natural language question answering, reasoning, etc.

Constructing KB's requires data to tap into. A crucial amount of human knowledge still resides in text documents such as books, articles, letters, news archives, and the Web documents. Thus, large scale KB's are constructed by extracting the knowledge from text and constituting it in a machine-understandable format. This task is called *knowledge base construction*.

KB construction methods extract information from natural language text by means of computational linguistics. These methods compile knowledge by parsing the unstructured elements of language, resolving the ambiguities, and linking these elements to canonical elements deliberately. In other words, transforming the lexico-syntactic constituents of the natural language to machine-readable canonicalized semantic constituents.



The research on KB construction has largely focused on populating KB's by extraction of i) new entities together with their semantic classes [2–5], and ii) new facts [6–31].

**1. Entity Extraction.** In the context of KB construction, entity extraction is the task of identifying the noun phrases that mention real world entities together with their semantic classes. To illustrate, entity extraction methods aim to detect phrases like “Ronaldo”, or “CR7”, and link them to the entity  $\langle \text{CristianoRonaldo} \rangle$  that is an instance of the semantic classes  $\langle \text{football\_player} \rangle$ , and  $\langle \text{person} \rangle$ .

**2. Fact Extraction.** Knowledge bases do not only contain entities and their classes, but also the relationships among them. The task of extracting the relationships between entities is called fact extraction (a.k.a. relation extraction). For example, the relation between the  $\langle \text{WorldPlayerOfTheYear} \rangle$  award and  $\langle \text{CristianoRonaldo} \rangle$  is a  $\langle \text{hasWonPrize} \rangle$  relation that could be stated through the fact  $\langle \text{CristianoRonaldo hasWonPrize WorldPlayerOfTheYear} \rangle$ . Recently, little research has been focused on enriching knowledge bases with temporal scope of facts, as knowledge about facts is highly ephemeral [32–40]. For example, the fact  $\langle \text{CristianoRonaldo playsFor ManchesterUnited} \rangle$  has a validity time from 2003 to 2009. It can formally be represented as a 4-tuple  $\langle \text{CristianoRonaldo hasWonPrize WorldPlayerOfTheYear} [2003, 2009] \rangle$ . The facts having a temporal scope are called *temporal facts*.

As a result of success in knowledge extraction methods, current KB's capture millions of entities such as people, companies, and movies, their semantic classes, and facts about them.

## 1.2 GOALS AND CHALLENGES

A common bottleneck of current KB's is the poor *coverage of temporal knowledge*. Extending KB's with temporal knowledge, aside from the recent work on scoping facts with time, has not been investigated thoroughly. First of all, event entities like sports finals, political scandals, or natural disasters are the first class citizens of knowledge bases. However, there is less research about extending KB's with emerging event entities. As a result, the knowledge about event entities is fairly limited. Secondly, the research on extracting alias names for entities has not paid attention to event entities. Similarly, extracting alias names for temporal facts has not received any attention at all. As a result, KB's have very low coverage of alias names for events and for facts, an essential kind of temporal knowledge. Thus, this dissertation moves forward the

state-of-the-art research in temporal knowledge extraction. It goes beyond finding the temporal context of facts. It proposes novel approaches for populating KB's with temporal knowledge by stating the following goals: i) populating KB's with new event entities together with their semantic classes and informative facts about them, and ii) populating KB's with alias names for events and temporal facts.

These goals require addressing the following challenges in the field:

**C1** **Extracting event entities.** Despite the advances in temporal knowledge harvesting methods, the coverage of long tail event entities is fairly limited. For example, current KB's contain an event like the *UEFA Champions League Final 2013*. However, none of them contain a subsequent event *Bayern Munich Triple*, referring to this team's winning of three championships a week later. The reason is that the state-of-the-art research focused on extracting event entities along with their semantic classes mostly from Wikipedia and similarly curated sources<sup>1</sup>. As a result, they ignored local or emerging events like *Bayern Munich Triple* that are too specific for Wikipedia yet would be desirable to include in a high-coverage knowledge base. Such specific events, on the contrary, are usually covered in news articles and in similar media. Therefore, going beyond Wikipedia articles is a worthwhile research challenge to populate KB's with event entities and their semantic classes. Chapter 3 tackles this challenge.

**C2** **Extracting informative facts about events.** Events are complex entities with certain properties, i.e., events have begin/end dates, take place at certain locations, have primary actors involved in, have sub-events, or follow up events. Each of these properties are represented through facts in KB's. For example, we would like to automatically capture the following facts about an event like the *UEFA Champions League Final 2013*:

```
<2013UEFACHampionsLeagueFinal happenedIn London>  
<2013UEFACHampionsLeagueFinal happenedOn 2013-05-25>  
<2013UEFACHampionsLeagueFinal hasWinningTeam BayernMunich>.
```

KB construction methods so far have extracted facts about events mostly from Wikipedia's infoboxes and categories. However, extracting such facts from news and similar media is a harder challenge. Chapter 3 tackles this problem.

---

<sup>1</sup>The *UEFA Champions League Final 2013* captured as an article in Wikipedia ([https://en.wikipedia.org/wiki/2013\\_UEFA\\_Champions\\_League\\_Final](https://en.wikipedia.org/wiki/2013_UEFA_Champions_League_Final)) so as in many knowledge bases.

- C3 Extracting alias names for events.** Event names can be phrased in many ways. To illustrate, the entity `<2014WinterOlympics>` can be phrased as “Olympic Games in Sochi”, “Games of the XXII Winter Olympiad”, or “first Winter Olympics hosted by Russia”. It is significant to note that events are usually phrased in natural language texts without proper capitalization. Therefore, knowledge harvesting methods mostly fail to extract alias names for event entities. The research in this line rather focused on relatively easier entities such as persons, organizations, and locations. As a result, current knowledge bases have low coverage of alias names for events. Chapter 4 tackles the challenge of extracting alias names of events even if they are long and not properly capitalized phrases.
- C4 Extracting alias names for facts.** As with entities, a phrase can also be an alias name for a fact. For example, the phrase “the second term of Florentino Pérez” is an alias for the fact `<FlorentinoPérez holdsPosition PresidentOfRealMadrid [2009–now]>`. The task of extracting alias names of temporal facts has not been investigated at all. Populating knowledge bases with this kind of information is invaluable in terms of semantic search, query understanding, summarization, and deep text analytics. It is particularly significant for *temporal tagging* tasks, as aliases for temporal facts help to annotate text with the temporal scope of facts. To illustrate, the sentence “Cristiano Ronaldo’s transfer to Real made him the most expensive football player of the year.” does not contain any temporal expressions. It is not clear when Cristiano Ronaldo was the most expensive player of the year. However, the phrase “Cristiano Ronaldo’s transfer to Real” has a unique interpretation given a background knowledge. It is an alias for the fact `<CristianoRonaldo playsFor RealMadrid [2009, now]>`. Thus, the phrase “Cristiano Ronaldo’s transfer to Real” can be annotated with the year 2009. We call such textual phrases that denote events or temporal facts *temponyms*. Chapter 4 tackles this challenge.

### 1.3 CONTRIBUTIONS

The dissertation makes the following contributions that tackle the particular challenges outlined above:

- 1. Populating knowledge bases with named event entities from news.** The first major contribution of this dissertation is to go beyond Wikipedia articles to harvest knowledge about the events in the long tail as well as brand-new events. It presents a

method for extracting events from news articles to populate a high quality knowledge base (C1). The method takes news articles from various sources as input. Different similarity measures among news articles are modeled in a multi-view graph. This graph is coarsened via a novel algorithm based on the information theoretic principle of minimum description length. Thus, the related news articles are grouped into a canonicalized representation of event entities. These event entities are labeled with fine grained semantic classes (C1). It also extracts relevant informative facts about these events (C2): i) begin/end dates of events, ii) relationships among events (sub-events, temporal order of events, etc.), and iii) people and organizations participating in events. The results of this work were presented at the CIKM 2014 [41] and at the WWW 2014 [42] conferences.

**2. Populating knowledge bases with temponyms.** The second major contribution of this dissertation is a method to populate KB's with temponyms which are phrases denoting named event entities (C3) or facts (C4). The presented method parses input text to extract temponym candidates together with their contextual cues such as entity mentions and time expressions. The method, then, leverages this rich set of features around a temponym candidate to map it to a fact or event in a KB through an Integer Linear Program (ILP). The ILP uses joint inference to produce high-quality mappings of the temponym phrases to a knowledge base, thus canonicalizing the representation of temponyms. The contributions here are twofold: i) the knowledge base is populated with temponyms, and ii) the temporal information about events and facts are propagated to temponym phrases in text. Hence, by solving this problem, we create added-value temporal and semantic markup of text documents. The results of this work were presented at the WWW 2016 [43] Conference and at the TempWeb 2016 Workshop [44].

This dissertation is based on the material published in the proceedings of the following conferences:

- CIKM 2014 [41]
- WWW 2014 [42]
- WWW 2016 [43]
- TempWeb 2016 [44]

## 1.4 DISSERTATION OUTLINE

The rest of this dissertation is organized as follows. Chapter 2 begins by introducing the general framework of knowledge bases, and knowledge base construction methods. It also presents the state-of-the-art for each of these topics. Then, it summarizes the history of information extraction methods. It ends with a review of challenges and research directions, building a base for the novelty of our methods by outlining the bottlenecks of the previous research. Chapter 3 presents the methods for populating KB's with fine-grained emerging events, along with detailed semantic typing and relationships among events, as well as important entities participating in events. It also demonstrates a system for searching and exploring a knowledge base of such events. Chapter 4 presents the methods for populating KB's with temponyms, thus, allowing comprehensive tagging of temponyms in text with temporal scopes via linking these temponyms to knowledge base events and facts. It also describes the integration of a wide range of temponyms to a temporal tagging software. Chapter 5 provides concluding remarks and outlines possible directions for future research.

# CHAPTER 2

## BACKGROUND & RELATED WORK

Recently, there is growing interest in knowledge harvesting. However, certain topics such as populating knowledge bases with events and temporal facts are not investigated well enough. This dissertation is focused on populating knowledge bases with this kind of temporal knowledge. This chapter introduces background and related work required for the rest of the dissertation. The first section introduces the basic concepts in knowledge bases. The second section explains the concepts in information extraction. The prominent projects related to this dissertation are also explained in the sections. The third section compares related tasks to the research goals of the dissertation. Finally, the chapter ends with a summary.

### 2.1 KNOWLEDGE BASE PRELIMINARIES

#### 2.1.1 Knowledge Base

**Definition 2.1.1 — knowledge base.** A knowledge base is a semantic database used for the collection and management of knowledge in  $c$ . It describes world's entities, their semantic properties, and their mutual relationships in the form of logical statements.

Knowledge bases can be manually compiled by domain experts. Examples are Cyc [45], WordNet [46], SNOMED [47], and more [48]. Alternatively, knowledge bases can be automatically constructed by information extraction techniques as

well. Examples are KnowItAll [49], ConceptNet [50], DBpedia [51], Freebase [52], NELL [10], WikiTaxonomy [53], and YAGO [4, 18].

### 2.1.2 Ontology

In philosophy, ontology is the study of what exists [54]. In computer science we define an ontology as follows:

**Definition 2.1.2 — ontology.** An ontology is the description of a domain of knowledge along with its properties, classes, and relationships by a formal language [55, 56].

Some authors call only the description of classes and relations the *ontology*, and refer to the instantiations of classes and relations as the *knowledge base* [48, 55]. In this present work, we use both of the terms in the same way. Ontologies formalize knowledge through specific languages. The ontology languages are close to first-order logic in terms of expressiveness. The most commonly used ontology languages are based on description logic. The World Wide Web Consortium suggests RDFS/OWL as ontology language. RDFS/OWL is based on the Resource Description Format (RDF), a standard knowledge representation model. RDF stores data in the form of subject-predicate-object triples. We call such triples *facts*. Some examples of facts (taken from the YAGO ontology) are in Table 2.1. Ontologies commonly consist of classes, entities, literals, and relations.

### 2.1.3 Classes

Ontologies define a type system (a taxonomy) to organize entities into equivalence classes. These classes may include common types such as `<entity>`, `<event>`, `<person>`, or `<location>`. They may include more specific types such as `<striker(football)>`, `<football_player>`, `<sports_competition>`, or `<match_final>`. These classes are organized into higher classes. RDFS/OWL language uses the `<rdfs:subClassOf>` predicate to describe the subsumption relation between classes. An example from Table 2.1 is `<football_player rdfs:subClassOf person>`.

### 2.1.4 Entities and Literals

Ontologies define individual entities specified by the domain knowledge.

**Definition 2.1.3 — entity.** An entity is the central semantic item of a knowledge base that is a uniquely identifiable thing such as  $\langle \text{CristianoRonaldo} \rangle$ ,  $\langle \text{RealMadrid} \rangle$ ,  $\langle \text{Milano} \rangle$ , or  $\langle \text{2016UEFACHampionsLeagueFinal} \rangle$ .

Ontologies provide classes in order to specify the type information for entities. RDF-S/OWL uses the  $\langle \text{rdf:type} \rangle$  predicate to describe the “instance of” relation between entities and classes. An example from Table 2.1 is  $\langle \text{CristianoRonaldo rdf:type striker(football)} \rangle$ .

Ontologies define notion of literals.

**Definition 2.1.4 — literal.** A literal is a string that represents a value.

For example, the string `"2016-06-01^^xsd:date"` represents a particular date, where  $\langle \text{xsd:date} \rangle$  is a data type. RDF specifies values of the same type into data types. Another example of literals are alias names of entities. For example, the literal `"CR7"` is an alias name for  $\langle \text{CristianoRonaldo} \rangle$ . Alias names are represented by RDFS/OWL’s  $\langle \text{rdfs:label} \rangle$  predicate. Examples can be seen in Table 2.1.

ID	Subject	Predicate	Object
#1	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{wasBornIn} \rangle$	$\langle \text{Funchal} \rangle$
#2	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{wasBornOn} \rangle$	<code>"1985-02-05^^xsd:date"</code>
#3	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{hasWonPrize} \rangle$	$\langle \text{WorldPlayerOfTheYear} \rangle$
#5	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{playsFor} \rangle$	$\langle \text{RealMadrid} \rangle$
#6	$\langle \text{RealMadrid} \rangle$	$\langle \text{owns} \rangle$	$\langle \text{BernabéuStadium} \rangle$
#7	$\langle \text{2014FIFAWorldCup} \rangle$	$\langle \text{startedOn} \rangle$	<code>"2014-06-12^^xsd:date"</code>
#7	$\langle \text{2014FIFAWorldCup} \rangle$	$\langle \text{endedOn} \rangle$	<code>"2014-07-13^^xsd:date"</code>
#8	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{rdf:type} \rangle$	$\langle \text{striker(football)} \rangle$
#9	$\langle \text{2014FIFAWorldCup} \rangle$	$\langle \text{rdf:type} \rangle$	$\langle \text{sports.competition} \rangle$
#10	$\langle \text{striker(football)} \rangle$	$\langle \text{rdfs:subClassOf} \rangle$	$\langle \text{football.player} \rangle$
#11	$\langle \text{football.player} \rangle$	$\langle \text{rdfs:subClassOf} \rangle$	$\langle \text{person} \rangle$
#12	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{rdfs:label} \rangle$	<code>"CR7"</code>
#13	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{rdfs:label} \rangle$	<code>"Ronaldo"</code>
#14	$\langle \text{CristianoRonaldo} \rangle$	$\langle \text{rdfs:label} \rangle$	<code>"dos SantosAveiro"</code>
#15	#3	$\langle \text{on} \rangle$	<code>"2008-##-##^^xsd:date"</code>
#16	#5	$\langle \text{since} \rangle$	<code>"2009-##-##^^xsd:date"</code>

TABLE 2.1: A fragment of the YAGO knowledge base in tabular form.



### 2.1.5 Relations

Ontologies define sets of predicates that capture interesting relations between knowledge base items such as entities, classes, or literals. The first argument of a relation is called *subject*, and the second argument is called *object*. The subject and the object stand in the relation given by the predicate. Thus, a predicate is the *indicator function* of a relation. In this present work, we use the term *relation* in the same way as *predicate*. Ontologies define domain and range restrictions for the subject and the object of predicates. Thus, each predicate has a type signature. For instance, the predicate  $\langle \text{hasWonPrize} \rangle$  has  $\langle \text{person} \rangle$  as its domain, and  $\langle \text{award} \rangle$  as its range.

Instances of predicates can be seen as the edges connecting semantic items, thus, creating subject-predicate-object (SPO) triples. A collection of such triples constitutes a *knowledge graph*. Such graphs have great benefits. Applications include semantic search over entities and relations, and natural language question answering over facts.

The SPO model provides many advantages. It offers flexibility to easily add new information to a knowledge base. New predicates can be defined with domain and range specifications. New information, then, can be added to the knowledge base by populating these predicates. This is how knowledge bases are populated with new knowledge. Another advantage of the triple model is the power it offers for higher order relations. A higher order relation takes a triple as its subject. In order to append additional information to a triple, RDF uses the *reification* mechanism. Reification works as follows: Assume that we would like to specify the validity time for a fact  $f$ . To express this constellation in RDF, an ID for  $f$  is created. Next, a second fact is created having the ID of  $f$  as its subject and the temporal information as its object. For example, given the fact  $\langle \text{CristianoRonaldo playsFor RealMadrid} \rangle$ , with the fact ID  $\langle \#5 \rangle$  the begin date of the validity of the fact (2009) is appended as  $\langle \#5 \text{ since "2009-##-##" }^{\wedge} \text{xsd:date} \rangle$ . Here, we assume that the predicate  $\langle \text{since} \rangle$  is defined upfront.

Ontologies contain different sorts of relations:

- *Unary relations*. These are the relations that denote the classes of sets of entities. The type information for entities in ontologies is represented through unary relations.

- *Binary relations.* These are the relations that hold between two entities (or between entities and literals) of particular types. An example is `<CristianoRonaldo wasBornIn Funchal>`.
- *N-ary relations.* Some facts may require more than just two arguments such as the facts with a time scope. These kind of facts are represented in RDF through the reification mechanism.

### 2.1.6 YAGO

A prominent project in the field of knowledge bases is YAGO. YAGO [4] is a knowledge base that combines Wikipedia and WordNet [46]. Each article in Wikipedia becomes an entity in YAGO. Particular categories in Wikipedia are used to extract type information for entities. YAGO uses a special algorithm to link this type information to the WordNet taxonomy [4]. This results in a very precise and large ontological information for entities. YAGO taps into Wikipedia infoboxes to extract facts about entities. YAGO has around 100 manually defined relations. Each has several extraction patterns to obtain facts. As a result, YAGO has more than 100 million facts with 95% precision.

In this work YAGO, is the particular knowledge base that we aim to populate with temporal knowledge.

## 2.2 TEMPORAL KNOWLEDGE

### 2.2.1 Time

#### **The Concept of Time.**

Time is the continued progress of existence and events in the past, present, and future regarded as a whole [57]. The concept of time has been a long debated issue in philosophy [58]. In our work, we take a pragmatic approach about time.

#### **Time Points and Intervals.**

A *time point* denotes the smallest time unit of fixed granularity. Ontologies store temporal information as literals through particular data types such as RDFS/OWL's `<xsd:date>`. A time point is assigned to a fact if the fact is an instantaneous event.

Time points could represent years, days, or seconds. The representation of the granularity of a time point is ontology dependent. YAGO, for example, denotes

time points typically with a resolution of days, but sometimes with a cruder resolution like years. Dates, in YAGO, are denoted in the standard calendar format "YYYY-MM-DD<sup>ⓧ</sup>xsd:date" (ISO 8601). If only the year is known, the dates are written in the form "YYYY-##-##<sup>ⓧ</sup>xsd:date" with # as a wildcard symbol.

A *time interval* is the part of the time axis bounded by two time points as begin and end. Knowledge bases assign facts a time interval if they have a duration with known begin or end dates.

### 2.2.2 Temporal Knowledge Representation

Temporal knowledge is an old AI topic [59]. Russel and Norvig [60] define temporal facts as fluents, i.e., facts whose validity is a function defined over time. There are different approaches to model this notion in the context of knowledge bases. A popular approach is reification. Another approach [61, 62] uses quadruples instead of RDF triples. The fourth component of a quadruple is used for the validity time of fluents. A third approach [18] uses subject-predicate-object-time-location-context (SPOTLX) tuples, where the fourth component is time.

Regardless of how time is formally represented, there are two central concepts in temporal knowledge bases: events and temporal facts.

#### Events.

Events are special entities that occur at a specific time and place (e.g., sports finals, battles, elections, etc.). Knowledge bases assign particular semantic classes to events. Every event happens during a time interval. If the start and end date of an event coincides, then it happens on a time point with respect to a certain temporal granularity such as a day.

#### Temporal Facts.

Some facts are time variant such as presidencies, marriages, and some are not such as birth and death dates. The temporal information is appended to time-invariant facts as their object. For example,  $\langle \text{CristianoRonaldo wasBornOn } "1985-02-05"<sup>ⓧ</sup>\text{xsd:date} \rangle$ . The time variant facts, similar to events, have a start and an end showing their validity times.

### 2.2.3 YAGO2

YAGO2 [18] is the only publicly available knowledge base in which entities, facts, and events are anchored in both time and space<sup>1</sup>. It is an extension of YAGO knowledge base that goes beyond the SPO triples by introducing the temporal (T), spatial (L), and context (X) dimensions. This results in SPOTLX tuples. It extracts temporal facts from Wikipedia infoboxes, categories, and lists through particular extraction rules. These are created by extending the YAGO fact extraction rules with new temporal information extraction rules. As a result, YAGO2 is a large time aware ontology with more than 30 million temporal facts.

## 2.3 INFORMATION EXTRACTION

Extracting semantic information from text to populate a knowledge base is called *knowledge harvesting*. Information Extraction (IE) is the main technology for tackling the goals of knowledge harvesting [27, 66]. IE refers to the automatic extraction of structured information from unstructured sources. With roots in the Natural Language Processing community, IE is now part of many different communities spanning machine learning, information retrieval, databases, and Web mining. This section explains the evolution of the IE field from its early days to recent projects showing its present significance.

### 2.3.1 History of Information Extraction

#### **Template Filling.**

Early systems defined *templates* for information extraction. Templates are linguistic patterns manually created by experts. They are used to capture the structure of the facts sought in a given domain. Each template consists of slots to be filled. The slots are usually a set of attribute-value pairs, with the values being text strings. Early IE tasks were centered around template filling tasks [67–69].

#### **Message Understanding Conferences.**

DARPA (Defense Advanced Research Projects Agency) financed a series of conferences to encourage the development of new and better methods of information extraction. These conferences were called Message Understanding Conferences

---

<sup>1</sup>There is also YAGO2s [63, 64], a refactored version of YAGO2 with a transparent and modular architecture, and YAGO3 [65] an extension of YAGO2s with multilingual facts and entities.

(MUC) [70–73]. MUC focused on extracting domain-specific information from the text to fill slots of prescribed templates. Typical slots were, for example, the cause, the agent, the time, and the place of an event. At the sixth conference (MUC-6) [72] the named entity recognition task was added. The task was to mark the name of an entity as person, location, organization, time, or quantity.

#### **Automatic Content Extraction.**

NIST (The National Institute of Standards and Technology) assembled a research program for developing advanced information extraction technologies succeeding MUC. The program was called Automatic Content Extraction (ACE). Unlike the task of identifying the names of entities as in MUC, ACE pursued the goal of detecting entities as target objects, and relations between them [74, 75]. Thus, two important fields of IE emerged: i) entity extraction, and ii) relation extraction.

### **2.3.2 Entity Extraction**

#### **Named Entities.**

Naming things to refer to their identities is a long-standing issue in philosophy [76–79]. In IE, names of entities are noun phrases comprising of one or a few tokens in natural language text. Examples are names of persons, organizations, locations, products, dates, monetary expressions, numbers, quantities, percentages, and geo-political entities. In knowledge bases, the popular form of entities are entities with proper noun phrases, so-called *named entities*. Examples of named entities are persons, organizations, locations, products, and geo-political entities.

#### **Named Entity Recognition.**

*Named entity recognition (NER)* is the task of identifying mentions of named entities in text and classifying them into pre-defined categories. These categories are a handful of coarse types such as persons, organizations, locations, temporal expressions, quantities, monetary values, and percentages. NER was first introduced in MUC-6 [72]. It consists of three subtasks:

- Recognition of proper names and acronyms of persons, locations, and organizations (ENAMEX),
- recognition of temporal expressions (TIMEX),
- recognition of monetary and numeric expressions (NUMEX).

The survey by Nadeau and Sekine [80] gives a comprehensive overview of the methods for named entity recognition.

One of the most widely known general-purpose NER systems is the *Stanford NER* tool [81]. Stanford NER is a Named Entity Recognition tool implemented in Java. It includes named entity recognizers for English and other languages, particularly for the three ENAMEX classes: PERSON, ORGANIZATION, and LOCATION. Stanford NER uses Conditional Random Field (CRF) classifiers [82] to label entities with pre-defined classes.

### **Named Entity Disambiguation.**

NER seeks for identifying the phrases denoting an entity, without knowing which unique entity is denoted. *Named entity disambiguation (NED)*, on the other hand, is the task of mapping ambiguous names in texts to unique entities in the real world. NED systems usually employ NER methods as the first step to recognize names of entities. The phrases recognized by NER are called *mentions* or *alias names* of entities. The goal of NED, then, is to link these mentions in text to canonical entities in a repository like a knowledge base. The dissertation by Hoffart [83] gives a detailed and comprehensive overview of the methods for named entity disambiguation.

Hoffart et. al. introduced the NED system called *AIDA* [84] which has become the state-of-the-art tool for NED. AIDA is a framework and online tool for entity detection and disambiguation. Given a natural-language text, AIDA extracts entity mentions and maps them onto canonical entities in a knowledge base like YAGO. AIDA exploits the similarity between the contexts of the entity mentions, and the coherence among candidate entities for the mentions. Thus, it jointly disambiguates mentions to canonical entities. Other prominent NED systems are [85–93].

### **Temporal Information Extraction.**

Temporal expressions are another type of entities that are introduced in MUC-6 [72]. The task was to merely recognize temporal expressions in text without classifying or disambiguating them. However, classifying temporal expressions and understanding their time value is crucial for many research tasks such as topic detection and tracking, document summarization, document retrieval, and fact extraction [39, 94, 95]. *Temporal information extraction* is the task of the i) extraction, ii) classification, iii) and normalization of temporal expressions occurring in text.

*Extraction* is the recognition of the phrases that are temporal expressions such as “14th of February 2015”, “last week”, “the next decades”.

*Classification* is about marking up temporal expressions with four pre-defined types: date expressions (e.g., Feb 14, 2015), time expressions (e.g., Wednesday 17:30), duration expressions (e.g., three decades), and set expressions (e.g., every Wednesday).

*Normalization* aims to map temporal expressions having the same semantics to the same canonical value in a standard format. For example, the temporal expressions “14th of February 2015” and “Valentine’s day of 2015” are normalized to “2015-02-14”. TIMEML is the standard specification language for temporal expressions [96]. The dissertation by Strötgen [97] gives a detailed and comprehensive overview of the field.

Strötgen et. al. developed the *HeidelTime* temporal tagger [98] which has a very high quality in terms of both precision and recall. *HeidelTime* is a rule-based system mainly using regular expression patterns for the extraction and normalization of temporal expressions. It uses linguistic clues for understanding the semantics of temporal expressions. Thus, it achieves the best results in classification and normalization of temporal expressions [99]. We used *HeidelTime* to extract and normalize temporal expressions throughout the work presented in this dissertation.

### 2.3.3 Relation Extraction

Once entities are extracted in text (including numbers, dates, etc.), the second step is to extract the relations that hold between these entities. This is called *relation extraction* (a.k.a. *fact extraction*). The output of relation extraction is usually in the form of subject-predicate-object (SPO) triples. Relation extraction systems typically fall into two categories according to whether the set of the relations are known apriori. These two categories are *open information extraction* [100] and *ontological fact extraction* [28]. Open information extraction (Open IE) does not assume the existence of entities or relations. Any noun phrase is considered to be an entity. Similarly, any phrase between entities is considered to be a relation. No semantics on the entities or on the relations is enforced. This results in high coverage but low levels of precision. Therefore, Open IE is not suitable to populate a high quality knowledge base. The methods introduced in this dissertation are to populate a knowledge base with temporal knowledge. Thus, our methods fall into the ontological fact extraction category.

#### **Ontological Fact Extraction.**

In contrast to Open IE, ontological fact extraction systems take the typed relations to be populated as input. One class of these systems extract facts by applying hand-crafted lexico-syntactic patterns over text documents. A popular example among these

are Hearst patterns [101] for extracting hyponym/hypernym pairs (i.e., instances of the  $\langle \text{type} \rangle$  relation). Many large-scale knowledge graphs are built through a similar process. Examples are YAGO, Freebase, and DBpedia. These projects extract facts from semi-structured sources such as Wikipedia via hand-crafted extraction rules.

Manually specifying extraction patterns yields highly precise facts. However, it does not scale when the number of relations increases. To deal with this issue, extraction systems that can automatically learn extraction patterns have been developed. These systems take relations and a few example facts for these relations as input. As a first step, these facts are spotted in text. Then, they automatically learn the patterns (textual phrases) for expressing the input relations. As a second step, they extract more facts by applying the learned patterns on text. Examples of these systems are DIPRE [8], Snowball [6], SOFIE [28], PROSPERA [24], and more [7, 9–23, 25–27, 29–31]. There are similar approaches to extract the temporal scope of facts. Recently, a sub-field of ontological fact extraction emerged for this goal, *temporal fact extraction*. It aims to populate a set of given relations with new facts with additional time annotations. There are various IE approaches that harvest temporal scope of facts [32–36, 38–40]. An example temporal fact extraction system is T-YAGO [33, 38]. This system extracts temporal facts given the relations taken from YAGO.

## 2.4 RELATED TASKS

This dissertation focuses on two tasks: i) populating knowledge bases with named events, ii) populating knowledge bases with temponyms which are phrases denoting named events or facts. Our work addresses these tasks from novel perspectives with novel contributions. We discuss how our methods differ from previous work below.

### 2.4.1 Populating Knowledge Bases with Events

The previous work on events falls into two categories: i) ontological event extraction, ii) mining events from news.

#### **Ontological Event Extraction.**

Knowledge bases such as YAGO [4, 18], DBpedia [51], or Freebase [52] contain entities of type  $\langle \text{event} \rangle$ . They contain fine-grained classes to which events belong, and facts about them. These projects extract events from curated sources such as Wikipedia. This yields very accurate output but low coverage. The reason is that they



capture a new event only after it has a sufficiently rich Wikipedia article. Thus, major events are captured only late, after being properly edited in Wikipedia. As a result, events in the long tail, and brand-new events are completely out of scope. However, we aim to capture more events of this kind as early possible. This is achieved by tapping into news articles rather than curated sources such as Wikipedia. Thus, we can populate a high quality knowledge base with a high coverage of named events extracted from news.

### **Story Mining from News.**

There is a large amount of work on topic detection and story mining on news [102–109]. However, the events found by these systems are not ontological events. An ontology requires canonical representations and semantic typing for event entities. In contrast, the output of previous work has neither of these. The goal of these systems was not to populate a knowledge base. They merely aim to group news of the same topic or story. Thus, the output is semantically not clean for populating a knowledge base. So there is a fundamental difference between traditional event mining and our goal of populating a knowledge base. Our methods combine the ideas of event mining from news, and ontological event extraction. We achieve semantically clean output with high precision and high coverage, by enforcing semantic constraints on the events extracted from news.

## **2.4.2 Populating Knowledge Bases with Temponyms**

Temponym detection and disambiguation has not been addressed in any prior work. To the best of our knowledge, this work is the first to introduce the task. However, there are several related research tasks.

### **Temporal Information Extraction.**

Temporal expressions in explicit, relative and implicit forms have been extensively studied as part of the TempEval competitions [99, 110]. Implicit temporal expressions (“Valentine’s day”, “new year’s eve”, etc.), in particular, are akin to temponyms. These expressions do not require a knowledge base of events or facts to be disambiguated, though. They can simply be resolved with hand crafted rules [111, 112]. In contrast, temponyms require to be mapped to knowledge base events or facts to be disambiguated. Moreover, temponyms are context dependent. We need methods that can leverage the context of a temponym (entities, temporal expressions, and other temponyms) to disambiguate it.

**Named Entity Recognition and Disambiguation.**

There are ample papers and software tools on named entity recognition and disambiguation (NERD) [113–115]. In particular, the survey by Shen et al. gives a comprehensive overview [115]. Methods for named entity recognition work well for names of people, places, and organizations. However, temponyms have a different nature than named entities, as they are phrased in text in many ways. As a result, general-purpose NERD tools often fail to recognize and disambiguate temponyms. The disambiguation of temponyms is a challenging task due to the semantic and temporal context that they depend on.

## 2.5 SUMMARY

This chapter defined central concepts in knowledge bases, including classes, entities, literals, and relations. It then discussed temporal knowledge and how it is represented. The chapter continued with a brief history of information extraction. It started with presenting the early IE tasks, proceeding to modern ontological fact extraction systems. Throughout the chapter, software tools and projects that are relevant to this dissertation have been introduced. Finally, the chapter discussed related tasks, by comparing them to our research goals.



# CHAPTER 3

## POPULATING KNOWLEDGE BASES WITH EVENTS

A *named event* is an entity that happened at a certain time or during a certain time period. Examples include battles, elections, concerts, championships, disasters, fairs, economic summits, etc. Named events are particularly important for knowledge bases. Modern knowledge bases contain entities of type `<event>` and fine-grained semantic classes which they belong. These knowledge bases contain facts about events such as temporal positioning and location of events, and involved entities. This knowledge enables a new semantic awareness in search, summarization, analytics, and recommendations. However, a problem of current knowledge bases is coverage; the amount of named events is fairly limited. The main reason is that entities and facts are mostly extracted from Wikipedia and similarly curated sources. None of the major knowledge bases taps into news or social media for increasing their population of named events. There is no methodology for harvesting named events in the long tail, and for capturing emerging events. This chapter presents this dissertation's contribution towards extracting named events from news to populate a high quality knowledge base.

### 3.1 MOTIVATION

**Named Event Extraction from Wikipedia.** Wikipedia based event extraction systems [4, 18, 51, 52] treat an event article in Wikipedia as an event entity. These

systems use regular expressions to extract facts from semi-structured elements (infobox, categories, lists, etc.) in each event article. This approach yields highly precise facts. However, they face the problem of low coverage. The major events, by this approach, are captured only late, after being properly edited in Wikipedia. Moreover, they cannot capture the dependencies between events such as subsequent events, or sub-events. For example, we would like to automatically capture an event like the *UEFA Champions League Final 2013*, along with its involved entities such as the winning team *Bayern Munich*. We also would like to capture a subsequent event *Bayern Munich Triple*, referring to this team's winning of three championships a week later. The *UEFA Champions League Final 2013* captured as an article in Wikipedia<sup>1</sup> so as in many knowledge bases. However, the *Bayern Munich Triple* is too specific for Wikipedia yet would be desirable to include in a high-coverage knowledge base.

**Event Mining from News.** Event and topic mining systems extract events and storylines from a batch or stream of news articles (e.g., [102, 103, 116]). These methods compute output at the level of (ambiguous) noun phrases such as news headlines rather than true event entities. These systems may extract high coverage of events but they lack on semantic quality. For example, a typical output of such a system could be events such as *Bayern wins Champions League*, *Bayern beats Borussia*, *Bayern's triumph in London*, etc. These are semantically overlapping entities, and do not denote distinct named events. Such simple clusters of news articles are not sufficient for a high-quality knowledge base. In contrast, we would like to reconcile all surface cues for the same event. For example, news articles with different titles, timestamps, entities would be reconciled into a single entity in a canonicalized representation. Moreover, we would like to assign fine-grained class labels such as `<soccer_finals>` or `<European_sports_championships>` to an event entity.

In summary, traditional event mining systems and Wikipedia based event extraction systems do not satisfy the goal of populating a knowledge base with fine-grained emerging events. Our goal is to overcome the key limitations of these systems by building a system with the following properties:

**1. Canonicalized event entities:** Event entities should be in canonicalized forms rather than mere clusters of text snippets or noun phrases from news articles. In other words, exactly all news referring to the same event should be placed in the same equivalence class. Besides, a representative name should be chosen for the event.

<sup>1</sup>[https://en.wikipedia.org/wiki/2013\\_UEFA\\_Champions\\_League\\_Final](https://en.wikipedia.org/wiki/2013_UEFA_Champions_League_Final)

**2. Semantic labeling:** Each event must be labeled with one or more semantic types from a knowledge base. In addition, events should be annotated with their participating entities like people, organizations, locations, etc. These in turn should also reside in the knowledge base, rather than being mere noun phrases.

**3. Temporal ordering:** Events must be placed on positions or intervals on the time axis. This is a requirement for chronological ordering of related events (e.g., one following the other, a semi-final match is followed by the final match, etc.), even if the news articles that constitute an event exhibit widely varying timestamps.

**4. Event hierarchies:** Events should be organized in a hierarchical manner, with refinements into sub-events and inclusion in super-events. For example, Edward Snowden's request for asylum in several European countries is a sub-event of the *Prism Scandal*, and *Arab Spring* has sub-events such as *Egyptian Revolution*, *Tunisian Uprising*, etc.

**Problem Statement.** Having the goals above in mind, we state the research problem of this work as follows:

- P** Populate a knowledge base with fine-grained emerging events, along with detailed semantic typing (e.g., using classes like rock concerts or hurricanes), relationships among events (sub-events, temporal order, etc.), as well as people and organizations participating in events.

The raw input for this task is articles from newspapers, magazines, and online feeds.

A seemingly obvious approach would be to run a clustering algorithm on a set of news articles, using a text-based content similarity measure. Each resulting cluster would then be interpreted as a separate event. Although, this general rationale is adequate, our goal is to go beyond this and to make the output as semantic as possible by having the four goals stated above. Thus, we can obtain semantically clear output that is ready for populating a high-quality knowledge base.

From a technical perspective, we define the following computational problem: Given a heterogeneous collection of news articles (from many different sources),

1. *group* these into equivalence classes denoting the same events,

2. *label* the equivalence classes with semantic types and participating entities,
3. *chain* them in chronological order on the timeline,
4. and *organize* them in a sub-/super-event hierarchy.

## 3.2 CONTRIBUTION

This work overcomes the key limitations of event extraction systems. The work contributes along the dimensions of the afore-mentioned four goals. It presents novel methods and software tools for populating event classes of high-quality knowledge bases. It does so by extracting, cleaning, and canonicalizing fine-grained named events from news corpora.

This work makes the following contributions:

1. mapping news articles into event classes by *automatically labeling* them with fine-grained types using statistical language models;
2. a multi-view graph model that captures relationships and relatedness measures between news articles;
3. a novel graph-coarsening algorithm for grouping and temporally ordering news articles based on the information-theoretic principle of minimum description length;
4. building a high-quality knowledge base with 25 000 named events automatically derived from 300 000 news articles;

An example output from the event knowledge base that we built is shown in Figure 3.1. Our method achieves this output, by processing several hundred articles about the UEFA Champions League season 2012/2013. The figure shows three different events and the news articles from which they were inferred. The representative names of the events are shown inside the ovals. The top band shows the semantic types of the events, and the bottom line shows their participating entities. The figure shows the positioning of the events on the time axis, and their chronological ordering. Note that the underlying articles in each cluster are not necessarily from the same date; so different clusters can overlap in their overall timespans. Inferring the temporal order between events is rather challenging. However, our method finds accurate temporal ordering between events in spite of this challenge.

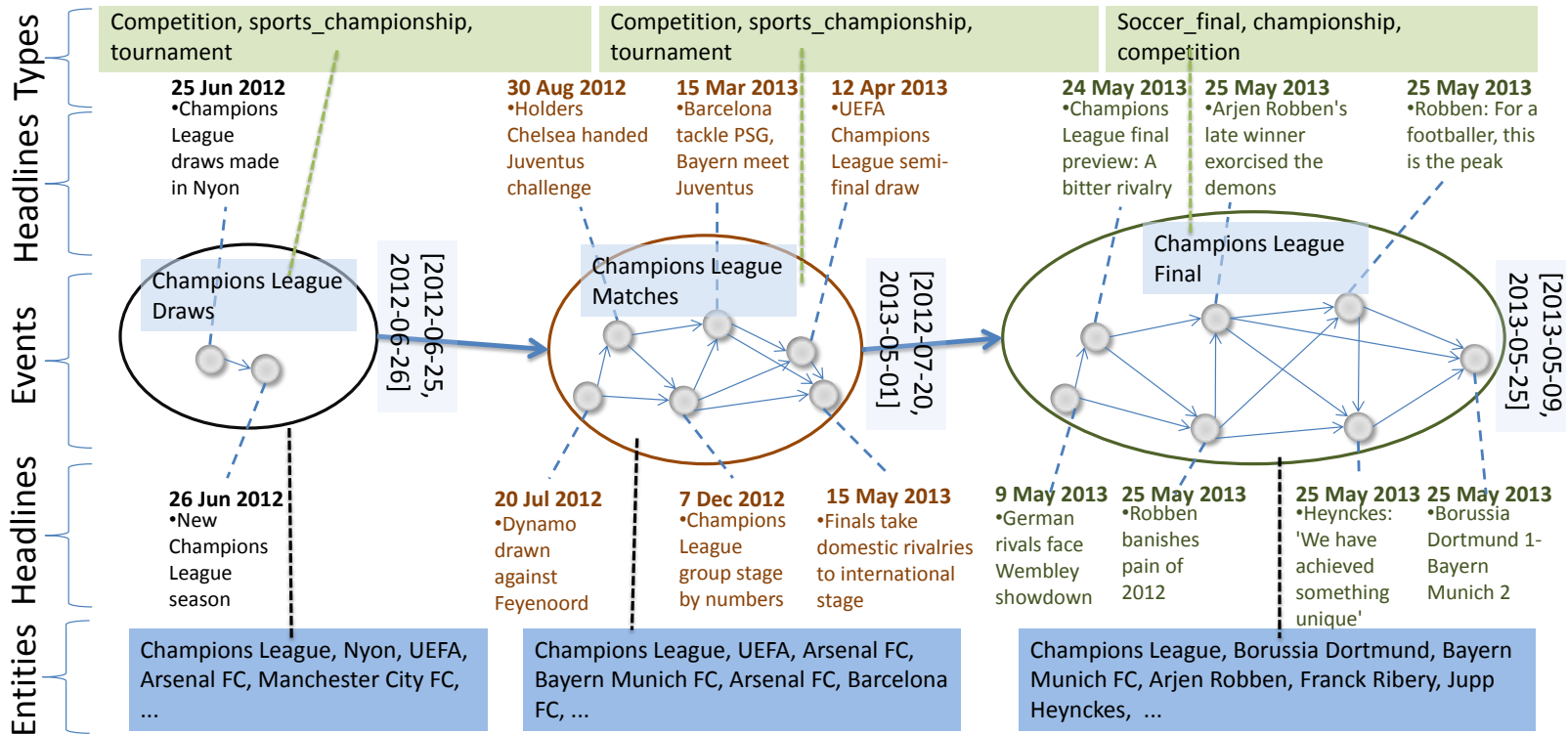


FIGURE 3.1: Output for the theme of “UEFA Champions League 2012/2013”.



<pre> {{Infobox Historical event   Event_Name = The French Revolution   Image_Name = Prise de la Bastille.jpg   Image_Caption = The storming of the Bastille, 14 July 1789   Participants = French society   Location = [[France]]   Date_start = 1789   Date_end = 1799   Result = [[Proclamation of the abolition of the monarchy]] }} ..... [[Category:French Revolution]] [[Category:18th-century rebellions]] [[Category:18th-century revolutions]] </pre>
---

TABLE 3.1: Wikipedia infobox and categories for the article French Revolution.

### 3.3 RELATED WORK

#### Ontological event extraction.

Knowledge bases such as YAGO [4, 18], DBpedia [51], or Freebase [52] contain entities of type event, fine-grained classes to which they belong, and facts about them. These systems extract events and facts about them from Wikipedia infoboxes and categories. Wikipedia infoboxes and categories are essentially typed records of attribute-value pairs that contain the most significant information about the event described by the article. For example, Table 3.1 shows the infobox and categories of the Wikipedia article about the French Revolution. The table contains information about the name of the event, its location, its date, the participants of the event, its semantic categories, and more. Therefore, a set of well-crafted regular expressions usually suffice to extract events. In addition, another rule-based method by [33] has shown how to increase the number of named events and temporal facts extracted from Wikipedia. The method does not only tap into Wikipedia infoboxes and categories, but also to lists, article titles, and other semi-structured elements in Wikipedia articles. However, all these approaches can extract an event and facts about it only if it has a sufficiently rich Wikipedia article (with infobox, categories, etc.). Capturing a brand new event or a local event in the long tail is out of hope for these approaches.

#### Event extraction in computational linguistics.

Another related work is event extraction task as introduced in the *TempEval Challenge* workshops [99, 110, 117], based on the TimeML markup language [96]. The task is

about identifying the temporal relations [118] between event-time and event-event. The state-of-the-art system on this task is TARSQI [37, 119] and TIE [34]. TARSQI detects temporal expressions by marking events and generating temporal associations between events. In the example sentence “Portugal won the UEFA Euro 2016 cup in France last month.”, systems like TARSQI consider the verb “won” to be an event, and the adverbial phrase “last month” to be a time point. Furthermore, they capture the relation between “won” and “last month”, in a so-called event-time relation. The TIE system does not only detect the temporal relationships between times and events, but also uses probabilistic inference to bound the time points of the begin and end of an event. In addition to TempEval tasks, there is also considerable work in NLP on events in narrative texts [120–123] and in clinical reports [104, 124].

The concept of event used in all these works differs fundamentally from our notion of ontological events. In contrast, they define events to be verb forms with tenses, adjectives, and nouns that describe the temporal aspects of the events.

#### **Event extraction and story mining from news.**

Historically news has been the best source of information on events. Thus, there is a large amount of work on event extraction and story mining on news, e.g., [102–105, 108, 116, 120, 125–129]. However, the events found by these systems are not ontological events. These events are computed at the level of (ambiguous) noun phrases such as news headlines rather than true event entities. For example, a typical output could be events such as “Real wins Champions League”, “Real beats Atletico again”, “Real’s triumph in Milan”, “Ronaldo’s decisive penalty for Real”, etc. These are semantically overlapping and do not denote distinct events. Simple clusters of news articles are not sufficient for a high-quality knowledge base. On the contrary, we require canonical representations and semantic typing of events. Thus, the output is clean enough for populating a knowledge base.

### **3.4 SYSTEM OVERVIEW**

We developed the EVIN system to extract events from news. The name is coined from the phrase “**EV**ents **I**n **N**ews”. EVIN takes a news corpus as input. This corpus can be obtained from online news from a heterogeneous set of sources such as newspapers, magazines, and other news feeds. A very first step is to extract basic features from this raw data by using information extraction tools. These features are the title, the content, the publication date, the entities, and the temporal expressions appearing in a

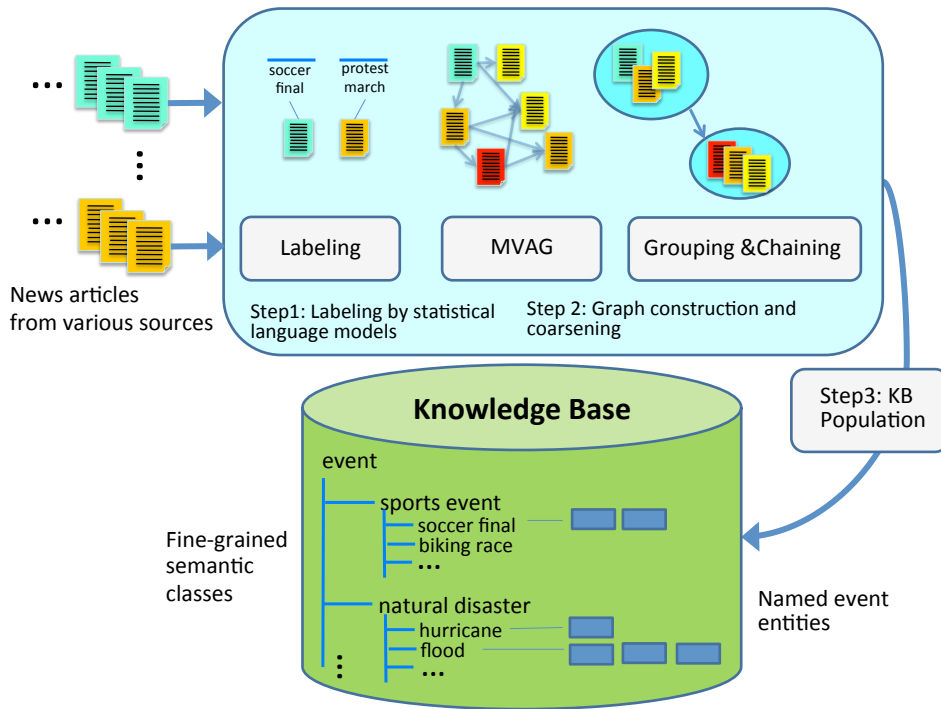


FIGURE 3.2: Architecture of the EVIN System.

news article. These are used to compute the various similarities between news articles. The feature sets and the distance measures to compute similarities are explained in Section 3.5. Later, extracting canonicalized and semantically organized event entities from news articles proceeds in three steps as shown in Figure 3.2:

**1. Labeling news with semantic types:** The first step is to label news articles to semantic event types in a knowledge base. The semantic labels of news articles are later carried over to the distilled events. The labeling phase is further detailed in Section 3.6.

**2. Grouping and chaining:** In this phase the feature sets (including semantic labels) and the distance measures are used together to construct a graph of news articles. The construction of the graph is explained in Section 3.7. News are grouped in a hierarchical manner based on coarsening the graph. Each group will eventually be an event. The coarsening step also takes care of chaining the related events in chronological order, Section 3.8.2. In order to identify the optimal coarsened graph, we developed an optimization model that employs information-theoretic principles, Section 3.8.3. We developed novel coarsening algorithms in the framework of our optimization model to extract events from a given MVAG, (Section 3.8.4).

**3. Populating a KB:** This step complements the output of the previous step by ranking the event candidates based on a scoring model. Thus, only the events with a high quality would be selected to populate a knowledge base. Finally, the events and their semantic annotations are placed in the knowledge base, populating a taxonomy of fine-grained semantic types. This is further detailed in Section 3.8.5.

## 3.5 FEATURES AND DISTANCE MEASURES

Our methodology extracts various features from news articles to compute various similarities.

### 3.5.1 Feature Sets

We exploit four kinds of feature groups that are provided by each news article  $n$  or can be derived from it by information extraction methods:

**Definition 3.5.1 — textual content.** It is the content of an article and denoted by  $\mathcal{C}(n)$  (or  $\mathcal{C}$  if clear from the context). Textual content consists of the words, the date literals, the numbers, appearing in the title and the body of a news article.

**Definition 3.5.2 — publication date.** It is the timestamp of a news article denoting the date of publication. The publication date of the news article  $n$  denoted by  $t(n)$ , (or  $t$  if clear from the context).

**Definition 3.5.3 — entity set.** The entity set of a news article contains the people, the organizations, the countries, the companies, etc., appearing in the textual content of the news article. The entity set of the news article  $n$  denoted by  $\mathcal{A}(n)$  (or  $\mathcal{A}$  if clear from context).

The entity names appearing in a news article are extracted using the Stanford NER tagger [81]. Please note that since these are used as features for subsequent processing, we do not attempt to disambiguate entities onto canonical representations in a knowledge base. We rather accept a tolerable level of ambiguity and noise.

**Definition 3.5.4 — semantic types.** Semantic types of a news article are the classes of events reported by the news article, for example, bombing, earthquake, festival, concert, etc. The semantic types of the news article  $n$  denoted by  $\mathcal{T}(n)$

(or  $\mathcal{T}$  if clear from context).

### 3.5.2 Distance Measures

Features are used to compute different kinds of distance measures between news articles.

**Definition 3.5.5 — content distance.** The content distance is the cosine distance of the articles'  $tf \cdot idf$  vectors over a bag-of-words model after stop-word removal:

$$dist_{text} = cosine(\mathcal{V}(n_i), \mathcal{V}(n_j))$$

where  $\mathcal{V}(n_i)$  and  $\mathcal{V}(n_j)$  are the  $tf \cdot idf$  vectors of news articles  $n_i$  and  $n_j$ , respectively;  $tf$  denotes the term frequency of words, and  $idf$  denotes the inverse document frequency of words (i.e., the inverse frequency of a word in the corpus).

**Definition 3.5.6 — temporal distance.** The temporal distance is the normalized time distance between the publication dates of news articles:

$$dist_{temp}(n_i, n_j) = \frac{|t(n_i) - t(n_j)|}{H}$$

where  $H$  is the time horizon of the entire news corpus. We define  $H$  as the difference between the earliest and the latest timestamps appearing in the corpus.

**Definition 3.5.7 — attribute distance.** The attribute distance between articles is the weighted Jaccard coefficient capturing the overlap in the *entity sets* or in the *type sets* respectively:

$$dist_{attr}(n_i, n_j) = \frac{\sum_{x \in \mathcal{X}(n_i) \cap \mathcal{X}(n_j)} weight(x)}{\sum_{x \in \mathcal{X}(n_i) \cup \mathcal{X}(n_j)} weight(x)}$$

where  $\mathcal{X}$  can be entity set  $\mathcal{A}$ , or type set  $\mathcal{T}$ . Entity names are weighted by their  $tf \cdot idf$  values, the types are weighted by their  $idf$  values.

These are all standard measures from the information-retrieval and text-mining literature. Our specific choices are based on prevalent usage in the state-of-the-art, but could be easily replaced.

Table 3.2 summarizes our notation in this work.

Symbol	Definition
$n$	A news item.
$\mathcal{T}$	Semantic types of a news item.
$\mathcal{A}$	Entities appearing in a news item.
$G$	A graph.
$V$	Vertex set of a graph.
$A$	Attributes of a graph.
$A(v)$	Attributes of the vertex $v$ .
$E$	Undirected edges of a graph.
$F$	Directed edges of a graph.
$dist_{text}(n_i, n_j)$	Content distance of two news items.
$dist_{temp}(n_i, n_j)$	Temporal distance of two news items.
$dist_{ent}(n_i, n_j)$	Weighted Jaccard distance between entity sets of two news items.
$dist_{type}(n_i, n_j)$	Weighted Jaccard distance between type sets of two news items.
$G^*$	A coarse graph.
$V^*$	Vertex set of a coarse graph.
$A^*$	Attributes of a coarse graph.
$E^*$	Undirected edges of a coarse graph.
$F^*$	Directed edges of a coarse graph.
$\Gamma$	Grouping function for mapping $V$ to $V^*$ .
rs	Resolution, the number of significant digits of the data.
$\mathcal{E}$	An extracted event.
$S(\mathcal{E})$	Score of an extracted event.

TABLE 3.2: Table of the notations used in this work.

### 3.6 COMPUTING SEMANTIC TYPES FOR NEWS

There are numerous studies on labeling documents with semantic categories like Wikipedia categories [130–132]. These methods exploit the semantic relatedness of documents to Wikipedia concepts and the link structure of Wikipedia categories. We have developed a method for mapping news articles onto semantic event types. News articles are mapped to event types in a knowledge base through specific Wikipedia categories for events. Our method works in two steps as shown in Figure 3.3.

First, news articles are mapped to Wikipedia categories (ca. 32 000 event categories). Second, Wikipedia categories are mapped to *Wordnet’s event classes* [46]. WordNet provides ca. 6 800 classes under the type label  $\langle event \rangle$  in its taxonomy, with 5-6 levels of subclasses. Examples are:

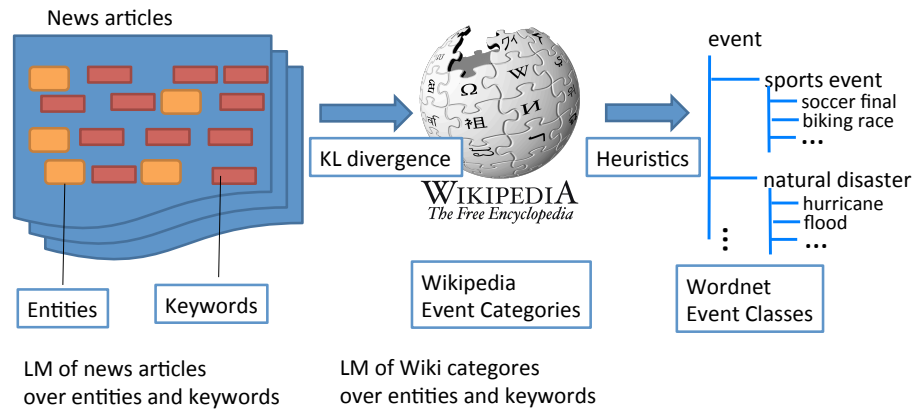


FIGURE 3.3: Two step of computing semantic types for news.

$\langle \text{final} \rangle \rightarrow \langle \text{match} \rangle \rightarrow \langle \text{contest} \rangle \rightarrow \langle \text{social\_event} \rangle \rightarrow \langle \text{event} \rangle$ , and  
 $\langle \text{riot} \rangle \rightarrow \langle \text{violence} \rangle \rightarrow \langle \text{aggression} \rangle \rightarrow \langle \text{action} \rangle \rightarrow \langle \text{act} \rangle \rightarrow \langle \text{event} \rangle$ .

The arrow symbol here denotes the *subClassOf* relation between the classes. All WordNet classes are also integrated in YAGO [4, 18], the specific knowledge base we aim to populate.

### 3.6.1 Mapping News onto Wikipedia Categories

Mapping news articles to Wikipedia categories is achieved in two steps: first, by constructing *statistical language models (LM's)* [133] for articles and for Wikipedia categories; second, by mapping an article to the most similar categories using *Kullback-Leibler divergence*. Before we present how news are mapped to Wikipedia categories, we give a short overview on language models.

**Definition 3.6.1 — language model.** A statistical language model (LM) is a probability distribution over sequences of words or other text features.

LM's are a principled approach in IR [133], widely used for query-result ranking, cross-lingual retrieval, question answering, etc.

LM's are usually defined for documents and for queries, with probabilistic [134] or information-theoretic distance measures [135] between LM's. The former uses the *query likelihood* model, whereas the latter uses the *Kullback-Leibler divergence (KL)* model. In the query likelihood approach, each document has a language model over terms. A query is then assumed to be generated from one of these language models.

In contrast, the Kullback-Leibler divergence model assumes that both the query and the document are two different language models. The similarity between these two LM's is then used for ranking. The advantage of this model over the query likelihood model is that it has an explicit language model for a query. Therefore, it is possible to extend this query language model with additional information. We prefer this model for our setting.

We formally define the Kullback-leibler divergence and explain how to construct the language models for news articles and for categories next.

### Language models for news and Wikipedia categories.

**Kullback-Leibler divergence:** To compare how similar a Wikipedia category is to a news article, we use the Kullback-Leibler divergence (aka. relative entropy) to measure the difference between the corresponding LM's [136]. This is a standard measure in IR and NLP to find the difference between two probability distributions as defined below:

**Definition 3.6.2 — Kullback-Leibler divergence.** Given two discrete probability distributions  $P$  and  $Q$ , the Kullback-Leibler divergence, denoted by  $KL(P||Q)$ , is the expectation of the logarithmic difference between the probabilities and computed as:  $KL(P||Q) = \sum_i P[i] \cdot \log \frac{P[i]}{Q[i]}$

In our setting, we compute the KL divergence between the language model of the news article  $LM_n$  and the language model of the category  $LM_c$  as below:

$$KL(LM_n || LM_c) = \sum_s P[s | n] \cdot \log \frac{P[s | n]}{P[s | c]}$$

where  $s$  is a term,  $P[s | n]$  and  $P[s | c]$  are conditional probabilities for  $s$  given the news or the category, respectively. The categories are then ranked in ascending order of their KL scores. For each news article, we compute the top-k categories based on this distance measure, and accept those categories whose KL divergence is below a specified threshold.

In order to use such a model for our setting, we must first construct the language model  $P[s]$  of a news article and a category:

**Constructing language models:** We customize and extend the notion of LM's as follows. The LM of a news article captures i) the words in news content including its title, and ii) all entities in the news article, including normalized date literals that appear



in the article. Analogously, the LM of a Wikipedia category captures i) the words in the articles belonging to that category, and ii) all entities in the articles belonging to that category, including normalized date literals that appear in the articles. Hence, we define the *document models* for a LM in our setting as follows:

**Definition 3.6.3 — document model.** The document model of a news article  $n$  is constructed over keywords and entities appearing in the article, so it is a probability distribution over  $\{w : w \in n\} \cup \{e : e \in n\}$  where  $w$  denotes keywords, and  $e$  denotes entities.

Check the example below.

■ **Example 3.6.1** The news article *Borussia Dortmund 1-2 Bayern Munich*<sup>a</sup>, starting with “Arjen Robben’s late winner exorcised the demons that have haunted him and Bayern Munich in the Champions League as they won a pulsating all-Bundesliga encounter ...”, has a document LM with:

- keywords: {winner, exorcised, demons, match, goal, ...}
- entities: {⟨ArjenRobben⟩, ⟨BayernMunich⟩, ⟨ChampionsLeague⟩, ⟨Bundesliga⟩, ⟨WembleyStadium⟩ ...}

<sup>a</sup><http://www.bbc.co.uk/sport/0/football/22540468>

The LM of a Wikipedia category is defined analogously and constructed from all Wikipedia articles that belong to the category.

**Computing LM probabilities:** The LM’s probability distribution for news article  $n$  is calculated as below:

$$P[s] = \mu P_W[s] + (1 - \mu) P_E[s]$$

where  $s$  is a word or entity,  $P_W[s]$  and  $P_E[s]$  are estimated probability distributions for words and entities in  $n$ , respectively, and  $\mu$  is a hyper-parameter that controls the relative influence of each of the two aspects. The LM of a Wikipedia category is computed analogously. Our scoring problem now narrows down to estimating the parameters of the language model of a news/category for every term (i.e., for every entity and word).

**Estimating LM parameters:** The LM parameters for both news articles and categories are estimated from frequencies in the underlying texts. However, the LM

parameters are not derived directly from the frequency counts, because a model derived this way is not robust against the terms that have not explicitly been seen before. Instead, some form of smoothing is required to assign a probability to unseen terms. We estimate the LM parameters with *Jelinek-Mercer smoothing* by using the global frequencies in the entire collection of all news articles and all categories, respectively:

$$P_W[s] = \lambda P[s | n] + (1 - \lambda)P[s | N]$$

where  $\lambda$  is the smoothing coefficient, and  $N$  is the entire collection of all news articles. We estimate  $P_E[s]$  for entities analogously.

### 3.6.2 Mapping Wikipedia Categories onto Wordnet Event Classes.

The second step of our two-step approach maps the accepted Wikipedia categories to their lowest (i.e., most specific) event type in the WordNet taxonomy. We adopt and adjust the heuristic of [4] to map Wikipedia event categories to Wordnet event types. This method (Algorithm 1) uses a natural-language noun-group parser on the category name to identify its head word, and maps the head word to the best matching WordNet type. If the head word is ambiguous, the word-sense frequency information of WordNet is used to make the choice. For example, for the category name “*General elections in Italy*”, the word “*elections*” is found as the head word. Wordnet contains four possible senses for the word “*election*” ranked by the word-sense frequency:

1. **election#1:** a vote to select the winner of a position or political office. “The results of the election will be announced tonight”
2. **election#2:** the act of selecting someone or something; the exercise of deliberate choice. “Her election of medicine as a profession”
3. **election#3:** the status or fact of being elected. “They celebrated his election”
4. **election#4:** the predestination of some individuals as objects of divine mercy (especially as conceived by Calvinists). “God’s election is an unconditional act of free grace that was given through his Son Jesus before the world began”

In this case the word “*election*” is mapped to the first sense: election#1: a vote to select the winner of a position or political office<sup>2</sup>.

<sup>2</sup><http://wordnetweb.princeton.edu/perl/webwn?s=election>

---

**Algorithm 1** WIKITOWORDNET(Wikipedia category  $c$ , WordNetEvents synsets)

---

```

1:  $head \leftarrow headCompound(c)$ 
2:  $pre \leftarrow preModifier(c)$ 
3:  $post \leftarrow postModifier(c)$ 
4:  $head \leftarrow stem(head)$ 
5: if there is a WordNet synset  $s$  for  $pre + head$  then
6:   return  $s$ 
7: if there are WordNet synsets  $s_1, \dots, s_n$  for  $head$  ordered by frequency then
8:   return  $s_1$ 

```

---

### 3.7 MULTI-VIEW ATTRIBUTED GRAPH (MVAG)

Identifying events in news streams has been addressed by modeling news articles as graphs of entities [103] or as graphs of keywords [106]. Both methods identify dense clusters in graphs. However, an event here is merely a set of temporally co-occurring entities and/or keywords. Thus, events are only implicitly represented. Moreover, these sort of plain graphs with one kind of edges cannot capture all the similarities between news articles along with their attributes. Although, attributed graphs have been used before [137, 138], they were always for purposes unrelated to our topic. In contrast, we aim for a graph model that captures all the afore-mentioned similarities between news articles along with their attributes such as entities, and semantic labels. Therefore, we devised a novel graph model by using the feature sets and the distance measures together to construct it. This graph is called *multi-view attributed graph* (MVAG).

**Definition 3.7.1 — multi-view attributed graph.** A multi-view graph is denoted as  $G = (V, A, E, F)$ , and consists of vertices  $V$ , directed edges  $E$ , undirected edges  $F$ , and attributes  $A$ .

We formally define the components of an MVAG below.

**Definition 3.7.2 — vertices.** A vertex  $v_i \in V$  of the MVAG corresponds to a news article  $n_i$ .

**Definition 3.7.3 — attributes.** Each vertex inherits a set of attributes  $A$  from  $n_i$ : its textual content  $\mathcal{C}$ , its timestamp  $t$ , its associated entities  $\mathcal{A}$ , and its types  $\mathcal{T}$ .

**Definition 3.7.4 — weighted edges.** The MVAG has two kinds of edges: undirected ones, edge set  $E$ , and directed ones, edge set  $F$ . All edges are weighted.

### 3.7.1 Construction of the Graph

Two vertices are connected by an undirected edge  $e_{i \leftrightarrow j}$  if they share at least one entity and at least one type:

**Formula 3.7.1 — undirected edges.**

$$\exists e_{i \leftrightarrow j} : \mathcal{A}(n_i) \cap \mathcal{A}(n_j) \neq \emptyset \wedge \mathcal{T}(n_i) \cap \mathcal{T}(n_j) \neq \emptyset$$

The weight of the edge is the content distance between two vertices;

**Formula 3.7.2 — weight of an undirected edge.**

$$w(e_{i \leftrightarrow j}) = \text{dist}_{\text{text}}(n_i, n_j)$$

Two vertices are connected by a directed edge  $f_{i \rightarrow j}$  if their timestamps indicate that they are ordered on the timeline. The weight of a directed edge is the temporal distance between the time stamps of vertices:

**Formula 3.7.3 — weight of a directed edge.**

$$w(f_{i \rightarrow j}) = \text{dist}_{\text{temp}}(n_i, n_j)$$

## 3.8 FROM NEWS TO EVENTS

### 3.8.1 Design Alternatives and Choice

Once we have all the features of news articles in a collection, including the semantic type labels, our goal is to distill canonicalized named events from this collection and organize the events into semantic classes of the knowledge base. This task entails a *grouping* and a *chaining problem*: combining thematically highly related news into a single event, and ordering the resulting groups along the timeline. (ideally even in a causal manner, but this is outside of this work).

#### Design Alternatives.

A straightforward approach to this problem would be to proceed in two phases: first

compute clusters of news articles based on similarity over all features, then infer ordering relations between clusters (e.g., by majority voting over the items in each pair of clusters). This straightforward approach has two drawbacks: First, the clustering itself would face the problem that there are no good cues about how many clusters we should ideally obtain. So this would call for hierarchical clustering or multilevel clustering for flexibility in determining the final events. But these methods are more expensive than flat clustering. Second, computing the ordering chains only after the clusters are determined may pose a poor if not unsolvable situation for the chaining step.

### **Our Choice.**

To overcome these issues, we designed a novel approach to this problem, by means of a *graph coarsening* which will be explained in Section 3.8.2. The rationale is that we transform the fine-grained MVAG for news articles into a coarser graph whose nodes correspond to the final event entities. Thus, our approach integrates the clustering and chaining tasks. For a principled method, we designed an optimization model based on information theoretic principles 3.8.3. In order to identify the optimal coarsened graph for a given MVAG, we developed coarsening algorithms founded on our optimization model 3.8.4.

## **3.8.2 MVAG Coarsening**

*Graph coarsening* is the task of creating structurally similar graphs to the (original) input graph but smaller ones. It does so by grouping vertices together and building condensed smaller graphs. Graph coarsening has been pursued by [139, 140] for plain graphs in the context of graph-cut algorithms. A related field to graph coarsening is *graph summarization* [141, 142]. Graphs can be summarized in terms of motif distributions such as frequent triangles or frequent subgraphs, or by showing aggregated views of a graph. The latter is related to our coarsening method. Another approach [143] summarizes an attributed graph in terms of  $k$  groups. The key differences to our setting are that this work considers only graphs with one kind of edges (as opposed to our notion of MVAG's) and that the user determines the number of groups, by the parameter  $k$ .

The graph coarsening approaches focused on plain graphs with one kind of edges so far. Our setting is different by being based on multi-view attributed graphs. We developed novel methods for graph coarsening with an optimization model specifically

designed for our problem of grouping and chaining the nodes of an MVAG. We define MVAG coarsening at a high level as below:

**Definition 3.8.1 — MVAG coarsening.** MVAG coarsening is the task of computing a smaller MVAG while keeping the main properties and the structure of the input graph.

We formally define the problem of MVAG coarsening as follows:

- P** Given a multi-view attributed graph  $G$  with node set  $V$ , weighted undirected edges  $E$ , weighted directed edges  $F$ , entity sets  $\mathcal{A}$ , and types  $\mathcal{T}$ , a coarser graph  $G^*$  with  $V^* \in 2^V$  (i.e., forming equivalence classes of nodes) and  $E^*, F^*, \mathcal{A}^*, \mathcal{T}^*$  is computed such that
- $G^*$  preserves the main properties and structure of  $G$ , and
  - $G^*$  is simpler (smaller, coarser) than  $G$ .

In order to compute a coarser graph, we define a grouping function that maps fine nodes to coarse nodes.

#### Grouping function.

The grouping of the original  $V$  nodes that leads to  $V^*$  should induce the undirected edges, directed edges, edge weights, and attribute sets of the coarsened graph. We achieve this by formally defining a grouping function.

**Definition 3.8.2 — grouping function.** The grouping function  $\Gamma$  maps  $V$  to  $V^*$  while inducing

- the edge sets  $E^*$  and  $F^*$ ,
- the edge weights in  $G^*$ , and
- the attribute sets  $A^*$  in  $G^*$

**Inducing edges:**  $\Gamma$  determines the edges between two coarse nodes if there is at least one edge between the finer nodes of the coarse nodes. Formally,

**Formula 3.8.1 — inducing edges.**

$$e_{\Gamma(x) \leftrightarrow \Gamma(y)} \in E^* \Leftrightarrow \exists e_{x \leftrightarrow y} \in E$$

$$f_{\Gamma(x) \rightarrow \Gamma(y)} \in F^* \Leftrightarrow \exists f_{x \rightarrow y} \in F$$

**Inducing edge weights:**  $\Gamma$  also determines the edge weights in  $G^*$  by averaging the weights of edges between all node pairs  $(x, y)$  that are mapped onto coarsened nodes  $(\Gamma(x), \Gamma(y))$ :

**Formula 3.8.2 — inducing edge weights.**

$$w(e_{x^* \leftrightarrow y^*}) = \text{avg}\{w(e_{x \leftrightarrow y}) \mid \Gamma(x) = x^*, \Gamma(y) = y^*\}$$

$$w(f_{x^* \rightarrow y^*}) = \text{avg}\{w(f_{x \rightarrow y}) \mid \Gamma(x) = x^*, \Gamma(y) = y^*\}$$

**Inducing attributes:**  $\Gamma$  induces entity sets  $\mathcal{A}^*$  in  $G^*$ . It is worth noting that the entity set of a node in  $G$  can be noisy due to imperfect quality of the named entity recognition tool. In addition, there can be entities mentioned in a news article that are not relevant to the reported event. For example, the entities  $\langle \text{BBCRadio} \rangle$ ,  $\langle \text{BBCSportwebsite} \rangle$ , extracted from the news article “*Champions League: Dortmund confident Mats Hummels will be fit*”, are not relevant to the event mentioned in the article. Hence, the grouping function  $\Gamma$  induces the entity set of  $x^*$  as the shared entities of the fine nodes that are mapped to  $x^*$ . Therefore, the main participants (entities) of an event are captured, whereas irrelevant entities are avoided.  $\Gamma$  induces types  $\mathcal{T}^*$  in  $G^*$  in the same way as entity sets, using the intersection of type sets.

Formally,

**Formula 3.8.3 — inducing attributes.**

$$A(x^*) = \bigcap A(x_i)$$

where  $\Gamma(x_i) = x^*$ , and  $A$  is the attribute set that can be either entities  $\mathcal{A}$  or types  $\mathcal{T}$ .

Figure 3.4 illustrates the MVAG coarsening.

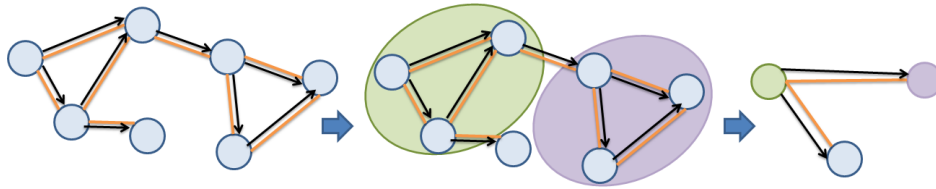


FIGURE 3.4: Coarsening a multi-view attributed graph.

### 3.8.3 Optimization Model

Our goal in graph coarsening is to create a graph i) as small as possible ii) while preserving the structure of the original graph. If a graph is coarsened too much, it would be very small but the structure would be lost. On the other hand, if a graph is not coarsened enough, it would preserve the original structure but it would not be as small as desired. Therefore, coarsening is a trade-off between how small the coarse graph should be and how much it should preserve the original structure. This reminds us the *law of parsimony* stated by the early philosophers like Ptolemy. He says:

“We consider it a good principle to explain the phenomena by the simplest hypothesis possible.” [144]

This rationale is in line with *model parsimony* or *minimum description length* in modern data mining. Both suggest that the best model in a family of models is the one that explains or predicts the data as much as possible with the fewest number of variables [145, 146]. In particular, the minimum description length (MDL) principle is a parameter-free approach with strong theoretical foundations. Hence, in order to identify the optimal coarsened graph for a given MVAG, we employ the MDL principle [147, 148].

#### Minimum Description Length.

In general, the MDL principle, is a practical version of Kolmogorov Complexity [149, 150]. The intuition behind MDL can be summarized by the following quote:

“Any regularity in a given set of data can be used to compress the data, i.e. to describe it using fewer symbols than needed to describe the data literally.”[148]



Indeed, the model that captures the most regularity in the data can achieve the best compression. Therefore, the MDL principle is often paraphrased as *Induction by compression*.

MDL is formally defined as follows.

**Definition 3.8.3 — minimum description length.** MDL finds the best model  $M^*$  in a family  $\mathcal{M}$  of models  $M, M \in \mathcal{M}$  as the model minimizing

$$L(M) + L(D | M)$$

where

- $L(M)$  is the length, in bits, of the description of  $M$ , and
- $L(D | M)$  the length, in bits, of the data given  $M$ .

This scheme ensures that  $M^*$  neither overfits nor is redundant—otherwise another  $M$  would minimize the sum. In our setting, the family  $\mathcal{M}$  of models is the family of MVAG's.

We formalize our goal of graph coarsening with the formulation of MDL as follows.

- P** For a given MVAG  $G$  with node set  $V$  with node attribute sets  $A$ , and both weighted undirected edges  $E$  and weighted directed edges  $F$ , our goal is to compute the MDL optimal coarsened graph  $G^*$  with node set  $V^*$ , weighted undirected edges  $E^*$ , weighted directed edges  $F^*$ , attribute sets  $A^*$  such that the total encoded length

$$L(G, G^*) = L(G^*) + L(G | G^*)$$

is minimal.

Hence, our algorithms aim to find a minimum of this *objective function*. It is important to note that to ensure fair comparison between models, MDL requires lossless descriptions. In our setting, this means we are required to reconstruct the original MVAG without any loss, given the coarse graph.

There is previous work that uses the MDL principle for summarizing and understanding large graphs [151–153]. In particular, the work by Navlakha et.al. [154] is close to

the spirit of our graph coarsening. It addresses the task of lossy graph compression by means of summarization using the MDL principle. In contrast to our setting, these approaches are limited to only plain graphs. They do not consider graphs with attributes and with different kind of edges.

To use MDL in our work we have to define how we encode a model, i.e., a coarsened graph  $G^*$ , and how we encode the input graph  $G$  given  $G^*$ . For the latter the high-level idea is to encode the error of  $G^*$  with respect to  $G$ , i.e., we encode their exclusive OR,  $G^* \oplus G$ , such that the receiver can reconstruct the original graph without loss upon receiving  $G^*$  and  $G^* \oplus G$ . We formalize these encodings next.

#### Encoding the Coarse Graph, $L(G^*)$ .

To encode a coarse graph  $G^*$ , we encode all its properties:

1. its vertices,
2. the attributes of vertices,
3. the undirected edges, and
4. the directed edges.

Formally,

#### Formula 3.8.4 — encoding the coarse MVAG.

$$L(G^*) = L(V^*) + L(A^*) + L(E^*) + L(F^*)$$

Before we continue with encoding the vertices, the attributes, and the edges, we need to define how to encode the weights of the attributes and the edges.

**Encoding the weights:** The weights, in our setting, are real numbers that are in the interval  $(0, 1)$ , e.g.,  $0.234800$  or  $0.389007$ . In order to encode the weights we need to encode the *number of significant digits*. Significant digits indicate the measurement resolution of the data, which means all trailing and leading zeros are unimportant. Therefore, we define the concept of the *resolution* of the data.

**Definition 3.8.4 — resolution.** Resolution is the number of significant digits of the data. The resolution,  $rs$ , is calculated as:

$$rs = 10^{\#sign. digits}$$

Having defined how to encode the weights, we can proceed by encoding the vertices, the attributes, and edges as follows.

**1. Encoding the vertices:** Encoding the vertices entails encoding their number (upper bounded by  $|V|$ ), and encoding per vertex  $v \in V^*$  to how many and which nodes in  $V$  it maps. Hence,

**Formula 3.8.5 — encoding the vertices.**

$$L(V^*) = \log(|V|) + \log\left(\frac{|V| - 1}{|V^*| - 1}\right) + \sum_{v \in V^*} \log\left(\frac{|V|}{|v|}\right)$$

**2. Encoding the attributes:** We encode the vertex attributes by encoding the all properties of all attributes per vertex. Formally,

**Formula 3.8.6 — encoding the attributes.**

$$L(A^*) = \sum_{v \in V^*} \left( \log(|A^*|) + \log\left(\frac{|A|}{|A(v)|}\right) + \sum_{a \in A(v)} \log(rs) \right)$$

where  $A(v)$  is the set of attributes of  $v$ .

Hence, per coarse vertex  $v \in V^*$  we encode how many attributes it has, which these are, and their weights.

**3. Encoding the undirected edges:** For encoding the undirected edges we first encode their number by using the upper bound  $ub = |V^*|(|V^*| - 1)$ . Secondly we identify which edges exist, and then encode their weights again by using resolution  $rs$ . Thus, we encode the undirected edges as below.

**Formula 3.8.7 — encoding the undirected edges.**

$$L(E^*) = \log(ub(E^*)) + \log\left(\frac{ub(E^*)}{|E^*|}\right) + |E^*| \log(rs)$$

**4. Encoding the directed edges:** Encoding the uni-directional edges is almost identical; in addition we only need to transmit their direction, for which we require one bit per edge,  $|F^*|$ . Thus,

**Formula 3.8.8** — encoding the directed edges.

$$L(F^*) = \log(ub(F^*)) + \log\left(\frac{ub(F^*)}{|F^*|}\right) + |F^*| \log(rs) + |F^*|$$

**Lossless Reconstruction of the Original MVAG,  $L(G | G^*)$ .**

To reconstruct the original graph given the coarse graph, we need to transmit all information needed to interpret the coarse graph, as well as correct any errors it makes with regard to the input data. That is, we need to correct all missing and superfluous edges, attributes, and their weights. At a high level we first need to transmit the vertices of the original graph, and the error for attributes and for edges. Thus, we have the following formula:

**Formula 3.8.9**  $L(G | G^*) = L(|V|) + L(A | A^*) + L(E | E^*) + L(F | F^*)$

As  $L(|V|)$  is constant for all models we can safely ignore it. We transmit all information needed to interpret the coarse graph as follows.

**1. Transmitting the error for attributes:** We transmit the error per node by encoding

1. the number of missing and superfluous attributes
2. which attributes these are
3. the correct weights of these attributes

Thus,

**Formula 3.8.10** — transmitting the error for attributes.

$$L(A|A^*) = \sum_{v \in V} \left( \log(|A \setminus A^*|) + \log\left(\frac{|A \setminus A^*|}{|A(v) \setminus A^*|}\right) + \sum_{a \in A(v)} \log(rs) \right)$$

where  $A(v)$  is the set of attributes of  $v$ .

We encode the attribute weight errors using  $\log(rs)$  bits each—if one is willing to make assumptions on the error distribution other choices are warranted.

**2. Transmitting the error for undirected edges:** To reconstruct adjacency matrix  $E$ , we transmit the edges in the (upper diagonal part) of the error matrix  $E_c = E^* \oplus E$  where  $\oplus$  the exclusive OR. Thus,

**Formula 3.8.11** — transmitting the error for undirected edges.

$$L(E | E^*) = \log(ub(E)) + \log\left(\frac{ub(E)}{|E_c|}\right) + |E| \log(rs)$$

We encode the weight errors using  $\log(rs)$  bits per edge in  $E$ .

**3. Transmitting the error for directed edges:** Last, let us write  $F_c = F^* \oplus F$  for the error matrix for the uni-directional edges. As above, we define  $L(F | F^*)$  analogue to  $L(E | E^*)$ , but in addition need to specify the direction of edges in  $F$  which are unmodelled by  $F^*$  using one bit per edge. Hence, we have

**Formula 3.8.12** — transmitting the error for directed edges.

$$L(F | F^*) = \log(|F \setminus F^*|) + \log(ub(F)) + \log\left(\frac{ub(F)}{|F_c|}\right) + |F| \log(rs)$$

In sum, we now have a principled and **parameter-free** objective function for scoring the quality of coarse graphs for a multiview attributed graph.

### 3.8.4 MVAG Coarsening Algorithms

The coarsening algorithms we developed aim to minimize the objective function defined in the previous section for a given MVAG. As the problem is computationally expensive, we focused on approximation algorithms.

**R Remark.** For an input graph  $G$ , the set of all possible models  $\mathcal{M}$  is huge,  $2^{|V|}$ . Moreover, it does not exhibit any particular structure (e.g., sub-modularity) that we can exploit for efficient pruning of the search space. Hence, we resort to heuristics.

The algorithms proceed in iterations; each iteration aims to reduce the input MVAG by the following three operations; MATCH, MERGE, and UPDATE:

**Definition 3.8.5 — MATCH.** For a given node, the match operation finds a suitable node(s) to merge into a coarser node. The MDL based objective function matches the nodes that result in the largest gain.

**Definition 3.8.6 — MERGE.** This operation merges two or more nodes into a coarser node.

**Definition 3.8.7 — UPDATE.** Following a MERGE step the MVAG is updated to reflect the new edges, edge weights, attributes, and attribute weights that result from node merging.

Within this general framework, we developed two specific algorithms: a greedy method and a randomized method, which we explain next in turn.

### 3.8.4.1 The GREEDY Method.

A standard greedy method would have a MATCH operation that, for a given node, always chooses the node(s) for which a MERGE results in the largest gain of the objective function. While our method follows this general principle, it has a specific twist by performing a light-weight form of *look-ahead* on the options for subsequent iterations. Specifically, we determine in each iteration if the currently best merge can be combined with other merges whose node pairs overlap with the node pair of the best merge. Without this look-ahead, the algorithm produces many tiny event groups that are basically the same real event. This is obviously not desired. The look-ahead should not be used aggressively, otherwise it produces few huge event groups containing many different events. This is not desired, too.

The algorithm keeps track of the candidate node pairs considered for merging, using a priority queue  $Q$  of node pairs sorted in descending order of gain (line 2-3 in Algorithm 1). The gain is calculated as the objective function's improvement caused by a potential node pair merge. The algorithm first selects the pair at the head of the queue, which decreases the objective function the most. Next, in contrast to standard greedy techniques, our algorithm scans the queue for other pairs that have one overlapping node with the nodes of the head pair (line 6-7) — the look-ahead mechanism—. If such a pair is found, it is added to “*mergeSet*”. The algorithm then proceeds further and repeats considering further merges, until it exhausts a bounded part of the queue. The following example demonstrates the algorithm.

■ **Example 3.8.1** Suppose the algorithm found  $\langle n_1, n_2 \rangle$  as “*bestPair*”, and the next positions in the priority queue are  $\{\langle n_1, n_3 \rangle, \langle n_2, n_5 \rangle, \langle n_4, n_6 \rangle\}$ . We scan  $Q$  and identify  $n_5$  and  $n_3$  for merging as their best matches  $n_1$  and  $n_2$  are already included in “*mergeSet*”. The algorithm thus expands the “*mergeSet*” into the set

$\langle n_1, n_2, n_3, n_5 \rangle$  (line 7) and merges all these nodes in one step (line 8).

The subsequent UPDATE operation incrementally computes the necessary changes of the MVAG and updates all data structures for the graph. The gain values for nodes in  $Q$  are recomputed for the next level of coarsening. When there is no further coarsening operation that would improve the objective function, the resulting MVAG is returned.

---

**Algorithm 2** GREEDY(MVAG  $G$ )

---

```

1:  $Q \leftarrow \emptyset$ 
2: for all matched pairs  $(u, v) \in V$  with  $gain(u, v) > 0$  do
3:    $Q.insert((u, v), gain(u, v))$ 
4: while  $Q \neq \emptyset$  do ▷ Iterative coarsening phase
5:    $mergeSet \leftarrow Q.popHead()$ 
6:   for all overlapping pairs  $(n, m) \in Q$  with  $mergeSet$  do
7:      $mergeSet = mergeSet \cup \{n, m\}$ , remove  $(n, m)$  from  $Q$ 
8:   MERGE( $G, mergeSet$ ), UPDATE( $G$ )
9:    $recompute(Q)$ 
return  $G$ 

```

---

### 3.8.4.2 The RANDOMIZED Method.

Our randomized method is based on the principle of *simulated annealing* [155]. The name of this principle is inspired from the physical systems. Every physical system has an initial internal *energy* which is its state. The higher the energy, the more unstable is the system. Thus, the goal is to bring the system from an arbitrary initial state, to a state with the minimum possible energy by slowly *cooling down* the system.

Simulated annealing is an established framework for stochastic optimization, traversing the search space (of possible MERGE operations) in a randomized manner. The goal is to minimize the objective function (the energy of the system). In contrast to greedy methods, the method can accept, with a certain probability, choices that lead to worsening the objective function. The probability of accepting worse choices is gradually reduced (cooling down), so the algorithm is guaranteed to converge. Accepting worse solutions is a fundamental property of simulated annealing because it allows to escape from local optima.

The probability of accepting a merge operation is specified by an *acceptance probability function* which depends on the *gain* in the objective function and a on a global

time-varying parameter  $T$  called *temperature* (a standard notion in simulated annealing).

**Formula 3.8.13** — acceptance probability function.

$$P_A = e^{-\frac{\text{gain}(u,v)}{T}}$$

where  $u$  and  $v$  are the nodes to be merged.

Our algorithm, in each iteration, performs a randomized coarsening step. It does so by picking a random node  $u$  and then identifying its best MATCH  $v$  for a MERGE operation (line 3 in Algorithm 2). This is slightly different from standard methods where both nodes of a node pair would be chosen uniformly at random. Our experiments have shown that our choice leads to faster convergence. If the considered coarsening step decreases the objective function, it is accepted and the energy of the system is updated (line 5). Otherwise, it is accepted based on the acceptance probability function  $P_A$ . After each accepted coarsening step, the UPDATE operator adjusts the graph. The algorithm maintains the *temperature value*  $T$  which is gradually reduced after each iteration, by geometric progression with factor  $\alpha$ .

**R Remark.** Note that  $T$  is initially chosen very high and  $\alpha$  is chosen very close to 1, therefore, the temperature cools down very slowly after each iteration.

The algorithm terminates when the temperature drops below a specified threshold  $\varepsilon$ . Across all iterations, the best solution is remembered, and this is the final output when the algorithm terminates.

---

**Algorithm 3** RANDOMIZED(MVAG  $G$ ,  $\alpha$ ,  $\varepsilon$ ,  $T$ )

---

```

1:  $best \leftarrow \emptyset$ 
2: while  $T > \varepsilon$  do
3:   pick a random node  $u$ , and its match  $v$ 
4:   if  $\text{gain}(u,v) > 0$  then
5:     MERGE( $G$ ,  $\{u,v\}$ ), UPDATE( $G$ ),  $best \leftarrow G$ 
6:   else if  $\text{Random.probe} < e^{-\frac{\text{gain}(u,v)}{T}}$  then
7:     MERGE( $G$ ,  $\{u,v\}$ ), UPDATE( $G$ )
8:    $T \leftarrow T * \alpha$ 
return  $best$ 

```

---



### 3.8.5 KB Population with Events

The final step of our methodology is to populate a given KB with the extracted events. This step is realized in three stages:

1. Cleaning the extracted events
2. Choosing a representative name for events
3. Putting the events into the KB

#### 1. Cleaning the extracted events.

The set of extracted named events exhibit mixed quality and are not quite what a near-human-quality knowledge base would require. However, our framework for extracting events allows us to rank event candidates. We developed a *scoring model* to rank the event candidates. These candidates are then filtered by using thresholding for high precision.

**Scoring an event:** Our scoring model uses the *cumulative gain* for a coarse node representing an event in the output graph as a measure of quality.

**Definition 3.8.8 — cumulative gain.** Cumulative gain is the total improvement of the objective function after the merge operations that involved intermediate nodes that finally belong to the coarse node at hand.

Thus, the score for event  $\mathcal{E}$  is computed as:

**Formula 3.8.14 — event score.**

$$S(\mathcal{E}) = \sum_{q_i \in \mathcal{E}} \text{gain}(G, q_i),$$

where  $G$  is the graph,  $q_i$  is an intermediate node created by a merge operation that contributes to final output node  $\mathcal{E}$ .

**Thresholding:** The acceptance threshold for an event score is set to the 90th percentile of the score distribution for all extracted events. This is a conservative choice, motivated by the reported quality of 90 to 95% for knowledge bases like YAGO [18]<sup>3</sup>.

<sup>3</sup>Note that our task here is substantially harder than that of the YAGO extractors, as the latter operate on Wikipedia infoboxes.

- R** Thresholding may break chaining information between some events. Thus, chaining is re-computed by transitively connecting missing links.

## 2. Choosing a representative name for events.

Event names (rather than some event IDs) give us an insight about the event. Thus, each event should be labeled by a *representative headline*. The representative headline is chosen among the titles of the news that are in the same event group. For each event, the representative news article is chosen based on the *weighted degree centrality score* in the original MVAG. The title of the node with the highest degree centrality is chosen as the representative title of the event. The degree centrality is defined as:

**Definition 3.8.9 — degree centrality.** Degree centrality of a node  $n_i$  in an event group  $\mathcal{E}$  refers to the sum of the weights of the edges attached to  $n_i$  divided by the the sum of the weights of all edges in  $\mathcal{E}$ .

We compute the degree centrality over un-directed edges.

Predicate	Domain	Range
<code>&lt;hasNewsArticle&gt;</code>	<code>&lt;wordnet_event&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>
<code>&lt;hasParticipatingEntity&gt;</code>	<code>&lt;wordnet_event&gt;</code>	<code>&lt;owl:Thing&gt;</code>
<code>&lt;hasID&gt;</code>	<code>&lt;wordnet_event&gt;</code>	<code>&lt;xsd:integer&gt;</code>
<code>&lt;followedByEvent&gt;</code>	<code>&lt;wordnet_event&gt;</code>	<code>&lt;wordnet_event&gt;</code>
<code>&lt;subEventOf&gt;</code>	<code>&lt;wordnet_event&gt;</code>	<code>&lt;wordnet_event&gt;</code>
<code>&lt;hasPublicationDate&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>	<code>&lt;xsd:date&gt;</code>
<code>&lt;hasURL&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>	<code>&lt;yagoURL&gt;</code>
<code>&lt;reportingEvent&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>	<code>&lt;wordnet_event&gt;</code>
<code>&lt;hasTitle&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>	<code>&lt;xsd:string&gt;</code>
<code>&lt;hasContent&gt;</code>	<code>&lt;wordnet_news_article&gt;</code>	<code>&lt;xsd:string&gt;</code>

TABLE 3.3: The set of predicates to store events in YAGO.

## 3. Putting the events into the KB.

The particular KB we populate with events is YAGO. In order to store events in YAGO, we need to define a schema. An event is defined as an entity of class `<wordnet_event>`. Therefore, any sub-class of `<wordnet_event>` can be a type label for an event. Every news article is defined as an entity of class `<wordnet_news_article>`. We used existing YAGO predicates (relations) if applicable to our setting. Examples are `<startedOnDate>`, `<endedOnDate>`, `<skos:prefLabel>` (preferred label or name), and

`<rdf:type>`). Then, we define additional predicates with their domain and range specifications as shown in 3.3.

## 3.9 EVALUATION

First of all, we aim to evaluate the different components of our methods, in comparison with different baselines. Secondly, we would like to show the quality of the extracted named events from news through use cases. Finally, we aim to populate a high quality knowledge base with a high coverage of named events, which is our overriding goal in this work. Therefore, we designed three experiment sets for the three goals;

1. evaluating labeling, grouping, and chaining quality,
2. evaluating the quality of extracted events through use cases,
3. evaluating knowledge base coverage.

In each experiment set we present the setup, the evaluation task and metrics, and the results.

### 3.9.1 Experiment Set-1: Labeling, Grouping, Chaining Quality

#### 3.9.1.1 Setup

In this section, we present experiments to evaluate the output quality of the various components of our methodology, in comparison with different baselines. We do this systematically by looking at our three main components separately: labeling, grouping, and chaining.

**Datasets:** We prepared two test datasets: news articles from i) *Wikinews* and ii) news sources referenced in Wikipedia articles, *WildNews*. Note that Wikinews<sup>4</sup> is totally independent of Wikipedia. Moreover, we removed all semi-structured elements from Wikinews articles to create a plain text corpus. The Wikipedia-referenced news are a highly heterogeneous set, from a variety of newspapers and other online news providers. We, therefore, refer to this dataset as *WildNews*.

For the Wikinews dataset, we picked articles by starting from topic categories that denote named events. Such categories are identified by matching years in their titles,

---

<sup>4</sup>[en.wikinews.org](http://en.wikinews.org)

e.g., *FIFA World Cup 2006*, *War in South Ossetia (2008)*, *G8 Summit 2005*, etc. In total, we extracted around 70 named event categories from Wikinews, with a total of 800 articles. Some named events are simple such as *2010 Papal UK tour* or *Stanley Cup Playoffs 2007*, whereas others are complex and contain sub-events like *2010 Haiti earthquake*.

For the WildNews dataset, we start with the collection of Wikipedia articles listed in the *Wikipedia Current Events* portal<sup>5</sup>. All news items cited by these Wikipedia articles with external links are crawled and together constitute the news corpus. This corpus has 800 news for 26 named events.

**Ground Truth:** The way we derived the news articles from named event categories already provides us with ground truth regarding the grouping and chaining of articles. However, the data does not come with semantic labels for populating a knowledge base. To this end, we have randomly chosen a subset of 3 news articles per named event, a total of 210 articles for the Wikinews data and a total of 78 articles for the WildNews data. These samples were manually labeled with one or more semantic classes from WordNet taxonomy (which is integrated in the YAGO knowledge base). For example, the news article *Bomb blasts kill several in Iran* is labeled with  $\langle \text{wordnet\_bombing} \rangle$ ,  $\langle \text{wordnet\_death} \rangle$ ,  $\langle \text{wordnet\_conflict} \rangle$ .

### 3.9.1.2 Evaluation tasks and metrics

**1. Labeling.** Our methodology that uses statistical language models to find semantic labels of news articles is compared with the *tf · idf* based cosine similarity. As the labels computed by *tf · idf* cosine similarity and our LM-based method are ranked, we use the notion of *precision@k* to compare the *k* top-ranked labels against the ground-truth set of labels:

#### Formula 3.9.1 — Precision.

$$\text{precision@}k = \left( \sum_{j=1 \rightarrow k} r_j \right) / k$$

where *j* is the position, and  $r_j = 1$ , if the result at the  $j^{\text{th}}$  position is correct and  $r_j = 0$  otherwise.

We define *recall@k* analogously:

<sup>5</sup>[http://en.wikipedia.org/wiki/Portal:Current\\_events](http://en.wikipedia.org/wiki/Portal:Current_events)

**Formula 3.9.2 — Recall.**

$$recall@k = \left( \sum_{j=1 \rightarrow k} r_j \right) / n$$

where  $j$  is the position, and  $n$  is the number of ground-truth labels.

Although  $n$  is usually close to  $k$ , it can be smaller than  $k$  for certain news articles that have only few true labels. In this case, one may suspect that the models can easily reach 100% recall. However, in practice, none of the methods compared in this paper was able to reach 100% recall for the top-5 labels.

**2. Grouping.** We compare our Greedy and Randomized methods against several baselines: k-means clustering, hierarchical clustering, METIS [139] and the Storyline detection [108] method. All the baselines need a distance metric to compute the clusters of news articles. Therefore, all features (text, time, entities, types) are fed into the following flat distance measure that is used for the baselines:

**Formula 3.9.3 — Flat distance measure.**

$$d(n_i, n_j) = \alpha \cdot d_{text}(n_i, n_j) + \beta \cdot d_{time}(n_i, n_j) + \gamma \cdot d_{ent}(n_i, n_j) + \theta \cdot d_{type}(n_i, n_j)$$

The reported results in the experiments are obtained by uniform weighting. We varied these parameters to study their sensitivity. This led to small improvements (up to 5% in precision) in some cases and small losses in other cases. Hence, we only report results for the uniform setting. We explain these baselines next.

- **k-means clustering:** k-means clustering is a method of flat clustering that aims to partition data into  $k$  clusters. We ran k-means over both datasets with different  $k$  values. For each  $k$  value, the algorithm is run 10 times with different random centroid initializations. k-means achieved the best results, when  $k$  is set to the real number of the named events in the datasets. Thus, we set  $k = 70$  for the Wikinews,  $k = 26$  for the WildNews dataset.
- **Hierarchical agglomerative clustering (HAC):** Unlike flat clustering algorithms, HAC can find clusters with different levels of granularity. It is a method that seeks to build a hierarchy of clusters. We used an agglomerative strategy to find clusters of news articles. We tried different linkage criteria: single-link (SLINK) [156], complete link (CLINK) [157], unweighted pair group method average (UPGMA) [158], and weighted pair group method average (WPGMA)

[158]. WPGMA achieved the best results for our settings. Therefore, we only report the values for this algorithm.

- **METIS:** METIS [139] is a graph partitioning tool that first coarsens a graph and then applies partitioning algorithms to it. METIS takes  $k$  number of partitions as input. We incrementally changed  $k$  to see how clustering performance change. METIS can have the best result, similar to k-means, when  $k$  is close to the number of real clusters. Thus, we set  $k = 70$  for the Wikinews,  $k = 26$  for the WildNews dataset.
- **Storyline detection:** Storyline [108] is a method to find story chains in a set of news articles returned by a query. Representing news articles as nodes of a graph, the method applies two steps to compute chains: It first finds the minimum set of dominating nodes of the graph. Second, it defines the dominating node with the earliest time stamp as a root. Other dominating nodes are combined by a directed Steiner tree algorithm. The edge weights in the multiview graph are induced via the flat distance measure mentioned above. Note that Storyline detection is designed to return one story chain at a time. We modified the method to apply it repeatedly for each dominating node. Thus, it can find multiple chains in the test data without any querying.

To compare different methods for grouping news articles into events, we look at pairs of news articles. We have ground-truth statements stating which articles belong to the same group (i.e., named event) and which ones are in different groups. Thus, we can compare the output of different kinds of grouping algorithms against the ground truth. We define precision and recall as follows.

**Definition 3.9.1 — Precision.** The estimated probability of pairs  $(n_i, n_j)$  of news articles being in the same ground-truth event given that they are assigned to the same group.

**Definition 3.9.2 — Recall.** The estimated probability of pairs  $(n_i, n_j)$  of news articles being assigned to the same group given that they come from the same ground-truth event.

Let  $G_M$  and  $G_T$  denote the groups for method  $M$  or ground truth  $T$ , respectively. Then we compute (micro-averaged) precision and recall as:

**Formula 3.9.4 — Micro-averaged precision.**

$$precision = \frac{\sum_{G_M} |\{(n_i, n_j) \in G_M \mid \exists G_T : (n_i, n_j) \in G_T\}|}{\sum_{G_M} |\{(n_i, n_j) \in G_M\}|}$$

$$recall = \frac{\sum_{G_T} |\{(n_i, n_j) \in G_T \mid \exists G_M : (n_i, n_j) \in G_M\}|}{\sum_{G_T} |\{(n_i, n_j) \in G_T\}|}$$

**3. Chaining.** We compare the methods that can find dependencies between news articles. Among the models we introduced so far, Greedy, Randomized, and Storyline detection can find temporal dependencies. Thus, these are the only models used for chaining comparisons.

For evaluating the quality of ordering events on the timeline, we again consider pairs  $(n_i, n_j)$  of news articles, the ordering of their corresponding groups by method  $M$  and the ordering of their true events in the ground-truth data  $T$ . We refer to these orderings as  $C_M$  (chaining by  $M$ ) and  $C_T$  (chaining in  $T$ ). So  $(n_i, n_j) \in C_T$  means that there are ground-truth groups  $G_T, G'_T$  such that  $n_i \in G_T, n_j \in G'_T$  and  $G_T$  is connected to  $G'_T$  by a directed “happened-before” edge. We compute precision and recall analogous to grouping. Instead of using  $G_M$  and  $G_T$  groups for method  $M$  or ground truth  $T$ , we use  $C_M$  (chaining by  $M$ ) and  $C_T$  (chaining in  $T$ ) in the formulas.

**Formula 3.9.5 — Micro-averaged precision and recall.**

$$precision = \frac{\sum_{C_M} |\{(n_i, n_j) \in C_M \mid \exists C_T : (n_i, n_j) \in C_T\}|}{\sum_{C_M} |\{(n_i, n_j) \in C_M\}|}$$

$$recall = \frac{\sum_{C_T} |\{(n_i, n_j) \in C_T \mid \exists C_M : (n_i, n_j) \in C_M\}|}{\sum_{C_T} |\{(n_i, n_j) \in C_T\}|}$$

For all three tasks – labeling, grouping, chaining – our very primary goal is high precision as we aim to populate a high-quality knowledge base. Nonetheless, recall is also important for high coverage of events. The combined quality is usually measured by the harmonic mean of precision and recall, the F1 score:

**Formula 3.9.6 — F1 score.**

$$F1 = (2 * precision * recall) / (precision + recall)$$

k	Wikinews						WildNews					
	LM			Cosine			LM			Cosine		
	prec.	recall	F1	prec.	recall	F1	prec.	recall	F1	prec.	recall	F1
1	.71	.28	<b>.40</b>	.56	.21	.31	.59	.16	<b>.26</b>	.32	.09	.14
3	.58	.62	<b>.60</b>	.47	.50	.49	.54	.45	<b>.49</b>	.28	.22	.25
5	.58	.72	<b>.64</b>	.44	.62	.51	.47	.61	<b>.53</b>	.24	.30	.27

TABLE 3.4: Precision, recall, and F1 scores at  $k$ , for labeling.

### 3.9.1.3 Results

**1. Labeling results:** We compared our method to the baseline of using  $tf \cdot idf$  based cosine similarity. Both methods mapped the 78 WildNews and 210 Wikinews news articles, for which we had ground-truth labels, to Wordnet semantic classes through Wikipedia event categories (Section 3.6). The top-5 semantic classes of each of the two methods are compared against the ground-truth classes. Table 3.4 shows precision and recall for different ranking depth  $k$ .

We see that our method substantially outperforms the cosine-similarity baseline.

**2. Grouping results:** We ran flat clustering (k-means), multi-level clustering (METIS), hierarchical clustering (HAC), Storyline, Greedy, and Randomized on the Wikinews and the WildNews datasets. The final clusters produced by the methods are automatically evaluated by comparing them to the ground truth.

Although we gave k-means and METIS the “oracle” benefit of choosing  $k$  to be exactly the number of true events in the ground truth, they achieve inferior precision for both datasets. k-means obtained 62% precision, METIS 59%, and HAC 58% for the Wikinews dataset. For the WildNews dataset, k-means obtained 68% precision, METIS 71%, and HAC 50%. As precision is the crucial property for knowledge base population, k-means, METIS, and HAC are not suitable methods to populate knowledge bases for our setting.

The results for the remaining methods are shown in Table 3.5. Storyline prioritizes precision at the expense of poor recall. Although Storyline achieves a good precision for the WildNews dataset, it loses on F1 score due to low recall. Our methods, especially Greedy, yield similarly high precision at much higher recall. This results in more than doubling the F1 scores of Storyline.



	Wikinews			WildNews		
	prec.	recall	F1	prec.	recall	F1
Storyline	.79	.10	.18	<b>.91</b>	.15	.26
Greedy	<b>.80</b>	<b>.31</b>	<b>.45</b>	<b>.91</b>	<b>.38</b>	<b>.54</b>
Randomized	.79	.29	.42	.77	.26	.39

TABLE 3.5: Precision, recall, and F1 scores for grouping.

	Wikinews			WildNews		
	prec.	recall	F1	prec.	recall	F1
Storyline	.94	.32	.47	.96	.29	.45
Greedy	<b>.96</b>	<b>.77</b>	<b>.85</b>	<b>.97</b>	<b>.67</b>	<b>.79</b>
Randomized	.93	.71	.81	.94	.44	.60

TABLE 3.6: Precision, recall, and F1 scores for chaining.

**3. Chaining results:** We compare the methods that can find ordering dependencies between news articles. This rules out METIS, k-means, and HAC, and leaves us with the Storyline detection method as our competitor. As Table 3.5 shows, all methods achieve similar precision for chaining experiments for both datasets. However, Greedy and Randomized methods attain much higher F1 scores than Storyline.

#### 3.9.1.4 Discussion

We have three main observations on our Experiment Set-1:

**Observation 1:** The baselines METIS, k-means and HAC group news articles in an overly aggressive manner, resulting in clusters that unduly mix different events. Thus, they yield poor precision values. The Storyline method, on the other hand, favors pure clusters and is conservative in its chaining. This leads to low recall. In contrast, Greedy and Randomized methods exploit the rich features encoded in MVAG model well and jointly infer groups and chains, which results in high precision values and the best F1 scores for both datasets.

Considering that Randomized requires simulated annealing parameters as input, and those parameters may vary based on the news corpus, Greedy is the most practical parameter-free method with very good overall grouping and chaining performance.

**Observation 2:** All methods except Randomized perform well on grouping for the WildNews dataset, which seems surprising. The reason is that WildNews articles are longer than Wikinews articles and contain more entities. The average number of entities per article is 19 for Wikinews and 26 for WildNews. This observation suggests that the semantic features like entities in articles boost the grouping performance.

**Observation 3:** All methods perform better on the Wikinews dataset on chaining. The reason is that WildNews articles span a much shorter time period than Wikinews articles, and many articles have overlapping time stamps, which degrades the chaining performance. This observation implies that chaining is more difficult when news articles have nearby or overlapping timestamps.

### 3.9.2 Experiment Set-2: Quality of Events through Use Cases

#### 3.9.2.1 Setup

To evaluate the quality of extracted events we have formulated two questions:

- What is the quality of particular events, rather than overall (labeling, grouping, chaining) quality as explained in previous section?
- What is the quality of events belonging to particular semantic classes?

These questions naturally arise, if a user wants to search for a particular event, or for a class of events. This inspired us to develop a lightweight search engine that can search events for a query. This search engine is able to do *keyword search* and *semantic search*.

**EVIN search engine:** The events found by our experiments have 5 features; participating entities, semantic classes, start and end dates, a representative title, and the textual content of the news articles reporting the event. Therefore, we indexed the events by using these features. The particular events for the use cases are the events extracted from the WildNews dataset by the GREEDY method. We used the Lucene search engine<sup>6</sup> for indexing and searching. The system we created for this task is named *EVIN search engine*.

To search for events, two types of queries can be formulated, a keyword query, or a semantic query. For a query  $q$ , we run a multi-field boolean search over the different features of the events, retrieving a set  $S_{\mathcal{E}}$  of events:

---

<sup>6</sup><http://lucene.apache.org/>

**Formula 3.9.7 — Lucene search.**

$$S_{\mathcal{E}} = \{\mathcal{E}^i : \text{sim}_{\text{Lucene}}(q, \mathcal{E}^i) \geq \tau\}$$

where  $\text{sim}_{\text{Lucene}}$  is the similarity score of the boolean vector space model provided by Lucene and  $\tau$  is a specified threshold.

For our use cases we used  $\tau = 0$ , that means to return all the events that have any similarity to the query.

**3.9.2.2 Evaluation tasks and metrics**

We evaluated the quality of the extracted events based on the grouping and the chaining quality. The quality measure is precision as defined in Experiment Set-1.

**3.9.2.3 Results**

**Event quality for keyword search:** The WildNews dataset has 26 named events; we prepared 26 queries resembling each of the named events. For example, a user can find the events that *Egypt* involved via querying the system with the keyword “*Egypt*”. The quantitative evaluation for 10 of the 26 keyword queries are shown in Table 3.7. The table reports the number of named events, and the grouping/chaining quality returned for a given query. Moreover, the table reports the statistics for the average number of news articles in these events. For instance, for the query “*Egypt*”, five events are returned that are extracted from 95 news articles, an average of 19 articles per event. Note that if there is one event returned for a query, then the chaining quality is not computed. The results prove us that the quality for particular events is as good as the overall quality reported in Experiment Set-1.

**Event quality for semantic search:** We can also find the named events belonging to a certain semantic class. A user, for instance, can search for the all elections in the corpus via the semantic query “*wordnet\_election*”. We used top-5 most populated classes in the Wildnews dataset to query the events. The evaluation for the semantic queries are shown in Table 3.8. These results prove us that the quality for particular classes of events is as good as the overall quality.

Query	#events	Avg. event size	$P_{Gr}$	$P_{Ch}$
q1:“North Korea”	1	11	0.77	-
q2:“gun control”	2	10.5	0.75	0.98
q3:“huge protests”	2	7.5	1.0	1.0
q4:“incidents”	4	9	0.62	0.89
q5:“Italian elections”	1	6	1.0	-
q6:“meat scandal”	2	8.5	1.0	1.0
q7:“Academy awards”	1	16	0.87	-
q8:“Oscar awards”	1	13	1.0	-
q9:“school shooting”	3	23	0.97	1.0
q10:“Egypt”	5	19	0.96	0.97

TABLE 3.7: Examples of 10 keyword queries used for evaluating events.  $P_{Gr}$  stands for the grouping precision,  $P_{Ch}$  stands for the chaining precision

Query	#events	Avg. event size	$P_{Gr}$	$P_{Ch}$
q1:wordnet_bombing	4	6.5	0.73	0.95
q2:wordnet_protest	3	22	0.69	0.81
q3:wordnet_election	11	6	0.73	0.86
q4:wordnet_conflict	2	18.5	0.81	0.88
q5:wordnet_riot	10	17	0.79	0.91

TABLE 3.8: Examples of 5 semantic (wordnet) queries used for evaluating events.  $P_{Gr}$  stands for the grouping precision,  $P_{Ch}$  stands for the chaining precision

### 3.9.3 Experiment Set-3: Knowledge Base Coverage

#### 3.9.3.1 Setup

We performed a large-scale computation in order to populate a knowledge base with named events. We used the GREEDY method to process the news articles.

**Dataset:** We crawled 295 000 news articles from the external links of Wikipedia pages. This constitutes the news corpus we use for knowledge base coverage experiments. These news articles are from a highly heterogeneous set of newspapers and other online news providers (e.g., <http://aljazeera.net/>, <http://www.independent.co.uk>, <http://www.irishtimes.com>, etc.). Note that the Wikipedia pages themselves were not used.

### 3.9.3.2 Evaluation task

We show that the events distilled by our methods yield much higher coverage than traditional knowledge bases which solely tap into semi-structured elements of Wikipedia (infoboxes, category names, titles, lists). We compared two knowledge bases on their included events: YAGO [18], built from the 2013-09-04 dump of Wikipedia, vs. EVIN\_KB, the knowledge base compiled by our method as described above.

For comparing the two knowledge bases in their coverage of events, we used the *events* section of Wikipedia’s *year* articles as a form of ground truth. Specifically we used the pages <http://en.wikipedia.org/wiki/2012> and <http://en.wikipedia.org/wiki/2013>, but excluded events after 2013-09-04, the date of the Wikipedia dump for YAGO. In total, these articles contain 51 itemized snippets, each of which briefly describes a salient event. We also considered the month pages, e.g., [http://en.wikipedia.org/wiki/January\\_2013](http://en.wikipedia.org/wiki/January_2013), for the total relevant time period, together providing 9811 events in the same textual format.

Note that these text items are not canonicalized events: they mention an event but often do not point to an explicit Wikipedia article on the event, and in many cases such an explicit article does not exist at all. For example, the Wikipedia text for January 11, 2013 says: “The French military begins a five-month intervention into the Northern Mali conflict, targeting the militant Islamist Ansar Dine group.” However, the only href links to other Wikipedia articles (i.e., entity markup) here are about the French army, the Ansar Dine group, and the Northern Mali conflict in general. The event of the French intervention itself does not have a Wikipedia article.

For each of these textually described events in the *year* pages, we manually inspected YAGO and EVIN\_KB as to whether they cover the event as an explicit entity or not. For the 9811 event descriptions in the *month* pages, we inspected a random sample of 50.

### 3.9.3.3 Results.

**EVIN\_KB:** Our method computed 24 348 high-quality events from the underlying 295 000 news articles. A good portion of these events contain sub-events and follow-up events (i.e., events that appear on a temporal chain). Table 3.9 shows other statistics about the extracted events such as the number of distinct semantic classes that are populated with events, or the average number of entities participating per event, etc.

# of events	24348
# of sub-events	3926
# of follow-up events	9067
avg # of entities per event	18
# of distinct classes	453

TABLE 3.9: Statistics for extracted events.

Semantic Classes				
accident	acquisition	attack	award	battle
blackout	bombing	boycott	calamity	carnival
ceremony	championship	competition	concert	conclave
conflict	conquest	contest	controversy	crime
crisis	declaration	discovery	earthquake	election
film festival	final	flood	game	inauguration
insurgency	marathon	musical	opera	outbreak
play	political_movement	protest	rebellion	referendum
resolution	revolution	riot	scandal	sport_event
suicide bombing	tour	tournament	treaty	war

TABLE 3.10: Top-50 semantic classes populated with events.

	YAGO [2012-1-1,2013-9-4]	EVIN_KB [2012-1-1,2013-9-4]
total #	624	6423
% year events	16%	48%
% month events	10%	36%

TABLE 3.11: Coverage of events in YAGO vs. EVIN\_KB.

The most densely populated classes of events are: protest (8732), controversy (7514), conflict (4718), musical (1846), sport event (766), war (727), etc. An event can be assigned to multiple classes. The top 50 most populated classes are shown in Table 3.10

**Comparison to YAGO:** We compared the coverage of EVIN\_KB to YAGO. For both ground-truth sets (*year* and *month* pages of Wikipedia), the coverage figures are shown in Table 3.11. In both comparisons, EVIN\_KB shows more than 3 times higher coverage. This demonstrates the ability of our approach to populate a high-quality knowledge base with emerging and long-tail events. The total number of events

that EVIN\_KB acquired for the relevant time period is 10 times higher than what YAGO could extract from semistructured elements in Wikipedia. Table 3.12 shows two sample EVIN\_KB events that do not appear in YAGO, along with their semantic annotations, entities, and time spans.

<b>Q:</b> French military intervention in the Northern Mali conflict.			
<b>EVIN_KB:</b> France army in key Mali withdrawal			
<b>News</b>	<b>Entities</b>	<b>Classes</b>	<b>Time</b>
<ul style="list-style-type: none"> <li>• India pledges \$1 million to UN mission to Mali</li> <li>• Gunfire breaks out as Tuareg rebels enter city</li> <li>• France army in key Mali withdrawal . . .</li> </ul>	French Army, Mali, Tuareg, UN	conflict	[2013-01-11, 2013-05-25]
<b>Q:</b> Meteor explodes over the Russian city of Chelyabinsk.			
<b>EVIN_KB:</b> Russian Asteroid Strike			
<b>News</b>	<b>Entities</b>	<b>Classes</b>	<b>Time</b>
<ul style="list-style-type: none"> <li>• Russian Asteroid Strike</li> <li>• Exploding Meteorite Injures A Thousand People in Russia</li> <li>• UN reviewing asteroid impact threat . . .</li> </ul>	Russia, Chelyabinsk	death, disaster	[2013-02-15, 2013-02-17]

TABLE 3.12: Two sample EVIN\_KB events for given queries. YAGO has no results for these queries.



## 3.10 APPLICATIONS

The EVIN (**E**Vents **I**n **N**ews) project has developed methods and software tools for populating event classes (concerts, ceremonies, elections, conflicts, accidents, disasters, tournaments, etc.) of high-quality knowledge bases. This is done by extracting, cleaning, and canonicalizing fine-grained named events from news corpora. The EVIN\_KB presents a valuable resource for a variety of applications by its large and clean repository of events. We developed a Web-browser-based GUI for querying, exploring, and visualizing the events in the EVIN\_KB and their provenance. The system is called *EVIN browser*.

### 3.10.1 Implementation

The EVIN browser is implemented in Java, Javascript, CSS, and HTML.

For every event, the system finds representative images via querying a search engine based on the entities participating in the event. The relevance of the images to the event shows the quality of the coarsening, as coarsening can successfully induce the entities of an event.

Figure 3.5 shows a screenshot of the EVIN browser. The screenshot has three parts: the search and filter section at the top, the chaining band, and the grouping band. These serve the following purposes:

1. The system uses keyword/semantic queries to retrieve events from the populated knowledge base.
2. The chaining of events is shown at the chaining band.
3. The grouping of events is shown in the grouping band.
4. Upon clicking on an event, the lower band shows the semantic types, the entities, the images, and the news belonging to an event. The news items along with detailed info is also shown for the clicked event.

The EVIN browser system can be examined online at <http://www.mpi-inf.mpg.de/yago-naga/evin>.

**EVents In News Knowledge base browser** **EVIN-Demo**

You can browse the EVIN knowledge base by typing a key word or by writing the id of an event.

---

Semantic classes ▼

---

**Chaining**

```

graph TD
    E7[Event_7  
Wembley chosen to host 2013  
Champions League final] -- followedBy --> E107[Event_107  
Champions League draws made  
in Nyon]
    E107 -- followedBy --> E106[Event_106  
UEFA Champions League  
semi-final draw]
    E106 -- followedBy --> E104[Event_104  
Champions League Final  
Preview:  
Borussia Dortmund -  
Bayern Munich]
    E104 -- followedBy --> E102[Event_102  
Bayern Munich's  
Jupp Heynckes:  
'We have achieved  
something unique']
    E102 -- subEventOf --> E104
  
```

---

**Grouping**

[\[Expand All\]](#) | [\[Collapse All\]](#)

- + [Event\\_107: Champions League draws made in Nyon](#)
- + [Event\\_106: UEFA Champions League semi-final draw](#)

FIGURE 3.5: Screenshot of EVIN browser.

### 3.10.2 Scenarios

The system supports three ways of interactive search and exploration:



1. keyword-based querying,
2. navigation by semantic classes,
3. and combined search and exploration.

#### 1. Keyword-based querying.

Users type keyword queries such as “Champions League” to retrieve events matching

+ Event\_104: Champions League Final Preview

- Event\_102: Bayern Munich's Jupp Heynckes:

**Semantic**

**Classes:** tournament, final, championship, sport

**Entities:** Borussia Dortmund, Bayern Munich, Wembley, Franck Ribery, Robert Lewandowski, Arjen Robben

+ Bayern Munich's Jupp Heynckes: 'We have achieved something unique'

+ Robben: For a footballer, this is the peak

+ Bayern Munich 2 Borussia Dortmund 1: Robben banishes pain of 2012

+ Arjen Robben's late winner exorcised the demons

FIGURE 3.6: Grouping of the news into events.

the query. EVIN finds events like “*Event\_7: Wembly chosen to host 2013 Champions League Final*”, “*Event\_102: Bayern Munich’s victory*”, “*Event\_104: Champions League Final preview*”, and more. The first event includes news articles about the UEFA announcing the venue of the final, the second event includes Bayern Munich’s victory, the third event includes news from the week before the final. Figure 3.5 shows the chaining and sub-event information between these events. Figure 3.6 shows the grouping of the events and also news headlines as sub-items of the events. Detailed information can be viewed by clicking on items.

## 2. Navigating semantic classes.

EVIN uses WordNet event classes as semantic labels. The user can choose a semantic class from a drop-box menu in order to explore the events belonging to the class. For example, the semantic class `(wordnet_final)` contains all the events about cup finals.

## 3. Combined search and exploration.

By using the keyword-query box and the semantic-classes menu together, users can find events within a certain class. For example, the match finals between Bayern

Munich and Borussia Dortmund can be retrieved by combining the query “German rivals” and the semantic class `<wordnet.final>`.

### 3.11 SUMMARY

This chapter has addressed the goal of populating knowledge bases with named events extracted from news. The methods presented here fill an important gap in the scope and freshness of knowledge bases. We tap into (the latest) news articles for information about events, and distill the extracted cues into informative events along with temporal ordering. We do this by, first, mapping news articles into fine-grained event classes using statistical language models. Second, we introduced a multi-view graph model that captures relationships and relatedness measures between news articles. Third, a novel graph-coarsening algorithm for grouping and temporally ordering news articles based on the information-theoretic principle of minimum description length. Finally, we automatically extracted 25 000 named events from 300 000 news articles.

Our experiments demonstrated that our methods yield high quality, compared to a variety of baseline alternatives, and can indeed populate specific event classes in a knowledge base with substantial added value. Use-cases of this contribution include strengthening the entity-aware functionality of search engines, and also using the additionally acquired knowledge of events for smarter recommendations, e.g., when users browse social media, and for better summarization. The repository of events we built is freely available for interactive access and download at <http://www.mpi-inf.mpg.de/yago-naga/evin/>.

The events distilled by our methods are themselves entities serving as background knowledge for better acquiring more events and other temporal knowledge from Web sources. An important case is to use this repository to extract the temporal phrases that are alias names for events and temporal facts. In the next chapter we show how a system can extract and interpret such phrases by using a knowledge base of events and facts as background.



# CHAPTER 4

## POPULATING KNOWLEDGE BASES WITH TEMPNYMS

The previous chapter introduced EVIN, a system to populate knowledge bases with events extracted from news. Thus, it tackles the coverage problem of events in knowledge bases. Another important coverage problem is the alias names for the events and temporal facts in a knowledge base. In this chapter, we address this problem. We call the textual phrases that denote events or temporal facts *temponyms*. We present a method for comprehensive tagging of temponyms with temporal scopes via linking these phrases to knowledge base events and facts. Our method is based on an Integer Linear Program (ILP) and leverages time expressions and entity mentions in the context of temponyms. The ILP uses joint inference for high-quality mappings to a knowledge base. Detecting temponyms and mapping them to a knowledge base has not been addressed in any prior work.

### 4.1 MOTIVATION

Temporal expressions in text documents are important cues for searching information about events in web pages, news articles, and social media, and for analyzing historic perspectives over web and news archives [95, 159]. For example, a user searching for the “Maracana final 2014” should be shown information on the FIFA World Cup Final on July 14, 2014. The answers to a query on “summer festivals in Europe” should include the Roskilde Festival taking place in June/July, but should exclude the Tallinn Music Week taking place in March/April. Finally, a journalist or political analyst

looking for the “Alexis Tsipras inauguration” should obtain news, blogs, and user posts on the last Greek election which was on September 20, 2015, and a business analyst looking for the “market reaction to the Alibaba IPO” should see documents from September 18, 2014, or later.

To provide good answers to such time-oriented information needs, it is essential to extract and normalize *temporal expressions* – TempEx’s for short – in the underlying documents [160]. TempEx’s take different forms:

1. *Explicit temporal expressions* denote a precise time point or period such as “25-01-2015”, “Jan 25, 2015”, “01/25/15” “January 2015”, or “spring 2015”. All but the last one have a unique interpretation. The last expression can be normalized as well, by imposing a convention that the months of March, April and May count as spring.
2. *Relative temporal expressions* refer to dates that can be interpreted with respect to a reference date. Examples are “last week”, “next Monday”, “two days ago”, etc. The reference date is typically the publication date of a news article or user post.
3. *Implicit temporal expressions* refer to special kinds of named events that have a unique meaning, often of periodic nature, such as “Valentine’s day”, “Christmas”, etc.

Techniques for identifying and normalizing TempEx’s work well, although the normalization difficulty varies between the different forms of TempEx’s. It is significant to note the commonality between these forms; all of them are obvious temporal expressions. There is one kind of temporal phrases (e.g. “Alexis Tsipras inauguration” or “Clinton’s term as secretary of state”) that are not obvious temporal expressions. These *free-text temporal expressions* refer to arbitrary kinds of named events or temporal facts. Therefore, we define the fourth class of temporal expressions; temponyms.

**Definition 4.1.1 — temponym.** Temponyms refer to arbitrary kinds of named events or facts with temporal scopes that are merely given by a text phrase but have unique interpretations given the context and background knowledge about politics, sports, music, business, etc.

Examples of temponyms are “Roskilde festival”, “Greek referendum”, “Alibaba IPO”, “German triumph in Maracana”, “Clinton’s time as First Lady”, “second term of

---

Cristiano Ronaldo dos Santos Aveiro, (born *5 February 1985*), is a Portuguese professional footballer who plays for Spanish club Real Madrid and the Portugal national team. In *2008*, he won his first Ballon d’Or and FIFA World Player of the Year awards. Ronaldo joined Real Madrid during **the second term of Florentino Pèrez**. Since **his transfer to Real**, Ronaldo was the top scorer in the national league for the years *2011*, *2014* and *2015*. In **the World Cup in Brazil** his team Portugal left early. In the same year, Real was **the Champions League winner**, and Ronaldo was the **recipient of the Golden Ball** in *January 2015*.

---

TABLE 4.1: Excerpt from a biography of Cristiano Ronaldo.

Angela Merkel”, etc. We can see the differences between TempEx’s and temponyms by looking at the text in Table 4.1. The table shows an excerpt from a biography of Cristiano Ronaldo with explicit TempEx’s in italics and temponyms in bold.

As the definition of temponym says, they have unique interpretation given the context and background knowledge. Thus, we would like to find the ideal interpretations of temponyms by mapping them to a knowledge base of events and temporal facts. Figure 4.1 illustrates this scenario. The figure shows a text snippet about the football player Cristiano Ronaldo with temporal expressions highlighted and their ideal mappings to a knowledge base

Both events and facts have temporal scopes, in the form of time points when events happened or time spans during which facts hold. Our goal in this work is to detect the temponyms seen on the left side of Figure 4.1 and compute the correct mappings onto the right side. Note that all temponyms in the example correspond to events in time. However, some temponyms such as “Ronaldo’s transfer to Real” have to be mapped to general facts, such as  $\langle \text{CristianoRonaldo playsFor RealMadrid} \rangle$  rather than entities of type  $\langle \text{event} \rangle$ . One may argue that the temponym “Ronaldo’s transfer to Real” correspond to an event entity rather than a temporal fact. Here, we rely on how the knowledge base formalizes the phenomenon of “Ronaldo’s transfer to Real”; in the form of an event entity or a temporal fact. The knowledge base has freedom to choose among different representations. It may contain facts about stateful relationships without necessarily having explicit events for the begin and end of the relationships. Our methods aim to find the most ideal mapping for a temponym given the background knowledge base.



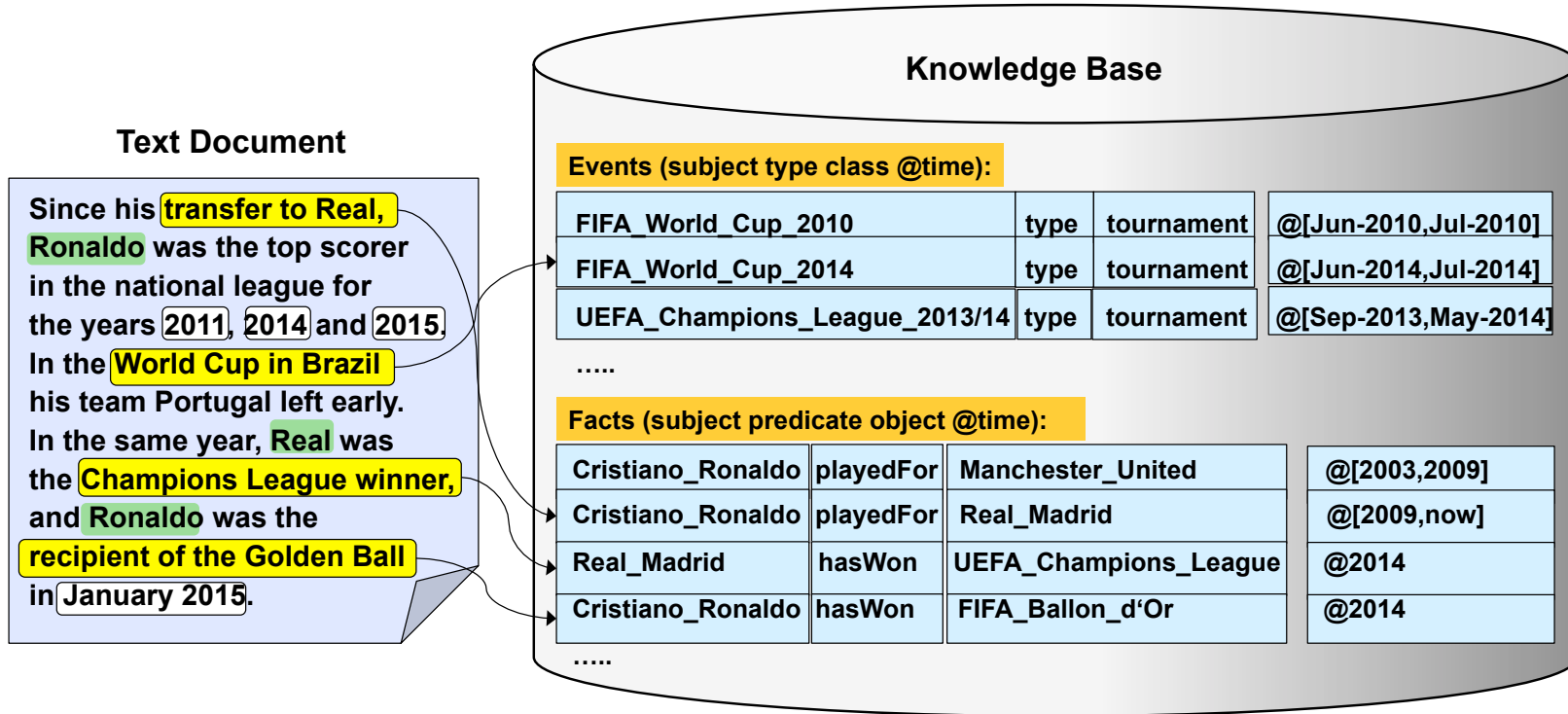


FIGURE 4.1: Example of Text with Temporal Expressions and Mappings to a Knowledge Base.

### 4.1.1 Challenges

#### **Limitations of State-of-the-Art Temporal Tagging.**

In recent years, good solutions have been developed for explicit, relative, and implicit dates. Most notably, tools like HeidelTime [98] and Tarsqi [119] perform TempEx extraction and normalization, and handle many such cases with very good precision and decent recall. However, there is no work on temponyms, which are addressed in this work.

#### **Limitations of State-of-the-Art Entity Linking.**

Temponyms about events are a special case of homonyms for individual entities: ambiguous phrases that denote people, places, companies, products, etc. Mapping names and phrases to entities in a data or knowledge base is known as *Entity Linking* or – more explicitly – as *Named Entity Recognition and Disambiguation (NERD)*, and there are ample papers and software tools on this task (see overviews like [113–115]).

Methods for NER, the recognition part, work well for names of people, place, organizations, and for explicit TempEx’s, but have poor recall for more sophisticated expressions of temporal nature like the ones considered in our work. General NED approaches for the disambiguation part produce fairly good results on people names. In addition, specialized solutions have been suggested for places, namely so-called *toponym* resolution approaches for geo-spatial entities [161, 162]<sup>1</sup>. However, the normalization of events and facts is more challenging and to the best of our knowledge, there is no prior work that specifically tackles the issue of extracting and disambiguating temponyms.

Note that general-purpose NED is inadequate for temponyms for two reasons. First, NED works well when it can exploit coherence (relatedness) measures between the candidate entities for different textual mentions. For example, knowing that the entities  $\langle \text{CristianoRonaldo} \rangle$  and  $\langle \text{RealMadrid} \rangle$  are highly related, helps jointly disambiguating “Ronaldo” and “Real”. However, the cues from the explicit TempEx’s that co-occur with a mention do not fall under this regime, because values like 2011, 2014, 2015 are not entities – so NED methods do not have a coherence measure between say 2015 and the entity  $\langle \text{FIFABallond’Or} \rangle$ . Second, our solution space includes mapping temponyms not just to entities of type event, but possibly also to entire subject-predicate-object facts such as  $\langle \text{CristianoRonaldo playsFor RealMadrid} \rangle$ . This is completely out of scope for NED.

---

<sup>1</sup>In analogy to toponyms, we coined the term temponym

### 4.1.2 Problem Statement

The problem addressed in this work is *temponym resolution*.

- P** Given a knowledge base of events with precise time points or periods as well as other entities, take as input an arbitrary text document from the web, news or social media, detect all temponyms, extract them, infer their temporal scopes, and map them to proper events or facts in the KB, thus canonicalizing their representation.

Especially, the mapping to facts is a demanding and novel approach that is beyond the scope of NED and has not been considered in prior work.

By solving this problem, we create added-value markup of text documents, which is a key asset for semantic search, query understanding, summarization, deep text analytics, KB curation, and other tasks. In addition to enriching input documents with links to the KB, temponym resolution also enhances the KB itself by providing additional alias names (aka. *paraphrases* or *mentions*) for known events.

## 4.2 APPROACH AND CONTRIBUTION

Similar to *named entity recognition and disambiguation* or *temporal expression extraction and normalization*, temponym resolution has two aspects: i) *temponym detection*, and ii) *temponym disambiguation*.

### 4.2.1 Temponym Detection

State-of-the-art TempEx taggers such as HeidelTime and SUTime are based on regular expression matching, handcrafted rules, and background dictionaries. For temponym detection in text documents, we adopt a similar approach and develop a rule-based system that uses similarity matching in a large dictionary of event names and known paraphrases. For example, in an input sentence like “Clinton served in the Obama administration”, the temponym “Obama administration” could be matched to a paraphrase of the event “presidency of Barack Obama” which has its own Wikipedia article and is an explicit entity in large knowledge bases. The input sentence “Beyoncé toured

with Jay-Z shortly after her marriage with him” is more challenging, as there is no dedicated article or entity on this marriage in Wikipedia or any knowledge base. In such cases, we attempt to match the temponym against subject-predicate-object facts in a knowledge base, like  $\langle \text{Beyoncé}, \text{isMarriedTo}, \text{Jay-Z} \rangle$  based on the cue that “marriage with” and “spouse of” are paraphrases of the same predicate. In knowledge bases like Freebase [52] or YAGO2 [18], such facts about relationships between two entities often have explicit time scopes that denote the validity time-points or timespans.

### 4.2.2 Temponym Disambiguation

Partial matching of temponyms with paraphrase dictionaries yields candidates for temponym resolution, but tends to produce a fairly noisy space of hypotheses. The next step – temponym disambiguation – is the key research challenge tackled in this work. To this end we harness the insight that temponyms co-occur with other TempEx’s and also with mentions of other entities involved in the event or fact to which the temponym should be mapped. By first resolving the simpler kinds of explicit, relative and implicit TempEx’s and by mapping co-occurring names of people, places, etc. to entities, we can create a rich set of features around a temponym and leverage these features for disambiguation. We further develop this idea into an *Integer Linear Program (ILP)*, whose solution yields good results yet can be efficiently computed on a per-temponym basis.

One limitation of this *local ILP* is that it does not consider other temponyms in the proximity nor does it take into account that the NED mapping for non-temporal entities is error-prone. Therefore, we improve this approach and devise a *joint ILP* that is aware of the NED uncertainty and computes a joint mapping of all temponyms and other entity mentions within a given document. Figure 4.1 illustrates this situation. Finally, an additional aspect to consider for joint inference is that the same temponym may occur in different documents. To leverage this richer context, we devise a *global ILP* that processes all temponyms and entity mentions of multiple documents simultaneously.

In summary, our contributions are

- the development of the first model for temponym resolution that uses joint inference for high-quality mappings to a knowledge base;

- tractable methods and a full system for enriching text documents by events and facts along with their semantic and temporal annotation
- populating a knowledge base with additional paraphrases of events and facts;
- comprehensive experiments with three corpora (biographies, history documents, news articles) that demonstrate the viability and quality of our solution.

### 4.3 PRIOR WORK AND BACKGROUND

#### **Temporal Expressions vs. Temponyms.**

To the best of our knowledge, temponym resolution, as we define it, has not been addressed in any prior work. However, there are several related research topics and we draw from some of their results as building blocks for our approach. Temporal expressions in explicit, relative and implicit form have been extensively studied as part of the TempEval competitions [160]. HeidelTime [98], SUTime [112], and Tarsqi [119] are some of the best performing systems that mostly rely on deterministic rules over regular expressions to perform both detection and normalization of TempEx's. The recent work of [163] pursues an alternative approach by learning context-dependent semantic parsers for TempEx's. None of this prior work addresses the case of temponyms.

#### **Event Extraction in Computational Linguistics.**

There is considerable work in NLP on events in narrative texts, based on the TimeML markup language [96], e.g., in “His first attempt to climb Everest was unsuccessful”, “to climb” is an event. Work along these lines includes [119–122, 164]. Recent work has further extended this direction to detect and align events in narrative texts using machine learning techniques [104, 124], with the specific target of clinical reports. Here events refer to the course of diseases and therapies of patients. The event definition used in all these works differs fundamentally from our notion of temponyms.

#### **Event Extraction in Web Mining.**

This work has focused on discovering events in news and generating storylines; see, e.g., [107, 165, 166]. However, the events found by these methods are not canonicalized and cannot be uniquely mapped to events in a knowledge base. Rather the output merely has the form of clusters of news articles or subgraphs of interrelated entities (involved in an event). Our work on populating knowledge bases with events as explained in previous chapter is an exception to that [41]. In that work, we map clusters of news articles into YAGO2 classes, thus assigning semantic types (e.g., sports

tournament, music festival, natural disaster, etc.) to newly found events. However, this method focuses on events that are not yet in the KB; it does not attempt to disambiguate clusters onto known events. Moreover, it relies on rich context from news articles where each article is about exactly one event; it is not designed to cope with short temponyms and the case where a document mentions many different events.

### KB's.

KB's are large repositories of individual entities like people, places, organizations, creative works (books, songs, etc.) and events, their memberships in semantic classes (aka. `<rdf:type>` or `instanceOf` predicate), and their attributes and relationships with other entities. Popular, publicly available KB's are DBpedia<sup>2</sup>, Freebase<sup>3</sup>, Wikidata<sup>4</sup>, and YAGO<sup>5</sup>. The contents of these KB's are in the form of subject-predicate-object (SPO) triples, following the RDF data model. For example, Hillary Clinton's position as secretary of state is captured by the triple `<HillaryClinton holdsPosition USSecretaryOfState>`, and her marriage has the form `<HillaryClinton isMarriedTo BillClinton>`. Events associated with time points are captured as entities with their respective types, for example: `<2014FIFAWorldCupFinal rdf:type football_tournament>`.

### Temporal Knowledge.

A few KB's, most notably Freebase and YAGO2, have augmented basic SPO facts by temporal (and also spatial) meta-facts. YAGO2 [18], which is used in our work, provides temporal scopes either by its `<happenedOn>` predicate, for example `<2014FIFAWorldCupFinal happenedOn 2014-07-13>`, or assigns time points or periods to reified facts. For example, for Clinton's term as secretary of state with fact id `f1` and for her marriage with fact id `f2`, the temporal meta-facts have the form: `<f1 validDuring [2009-01-21,2013-02-01]>` and `<f2 validDuring [1975-10-11,now]>`.

The YAGO2 methods for harvesting this temporal knowledge tap into infoboxes, category names, and lists of Wikipedia and use consistency reasoning for high-quality output [18, 33]. Examples of such *SPOT facts* (T for time) with their temporal scopes taken from the YAGO2 knowledge base are shown in Table 4.2.

<sup>2</sup><http://dbpedia.org>

<sup>3</sup><http://freebase.com>

<sup>4</sup><http://wikidata.org>

<sup>5</sup><http://yago-knowledge.org>

S	P	O	T
⟨FifaWorldCup2010⟩	⟨rdf:type⟩	⟨tournament⟩	⟨[2010-06-11,2010-07-11]⟩
⟨FifaWorldCup2010⟩	⟨startedOn⟩	⟨2010-06-11⟩	⟨[2010-06-11,2010-06-11]⟩
⟨FifaWorldCup2010⟩	⟨endedOn⟩	⟨2010-07-11⟩	⟨[2010-07-11,2010-07-11]⟩
⟨RealMadrid⟩	⟨won⟩	⟨UEFACHampionsLeague⟩	⟨[2014-05-24,2014-05-24]⟩
⟨RealMadrid⟩	⟨won⟩	⟨UEFACHampionsLeague⟩	⟨[2016-05-28,2016-05-28]⟩
⟨CristianoRonaldo⟩	⟨playsFor⟩	⟨ManchesterUnited⟩	⟨[2003-##-##,2009-##-##]⟩
⟨CristianoRonaldo⟩	⟨playsFor⟩	⟨RealMadrid⟩	⟨[2009-##-##,now]⟩
⟨CristianoRonaldo⟩	⟨hasWon⟩	⟨FIFABallond'Or⟩	⟨[2015-01-12,2015-01-12]⟩
⟨CristianoRonaldo⟩	⟨hasWon⟩	⟨FIFABallond'Or⟩	⟨[2014-01-13,2014-01-13]⟩
⟨AngelaMerkel⟩	⟨holdsPosition⟩	⟨GermanChancellor⟩	⟨[2005-11-20,now]⟩
⟨AngelaMerkel⟩	⟨holdsPosition⟩	⟨EnvironmentMinister⟩	⟨[1994-11-17,1998-10-26]⟩
⟨AngelaMerkel⟩	⟨bornIn⟩	⟨Hamburg⟩	⟨[1954-07-17,1954-07-17]⟩
⟨AngelaMerkel⟩	⟨bornOn⟩	⟨1954-07-17⟩	⟨[1954-07-17,1954-07-17]⟩

TABLE 4.2: Examples of SPOT facts.

Prior to that in [167] a temporal repository was created using Library of Congress subject headings. In [168] the original idea of gazetteer that includes geographical data was extended to named periods such as “Second World War”. To the best of our knowledge none of the existing works provide a repository of temponyms. Other methods for extracting temporal facts from text web sources or inferring the temporal scopes of known facts have been developed by [34, 35, 40, 169]. None of these machine-learning-based techniques has succeeded in scaling to large input and yielding high-quality output with precision above 90% and decent recall.

#### **Named Entity Recognition and Disambiguation.**

Named entity recognition and disambiguation (NER/NED) is the general task of detecting names and phrases that denote entities (NER) and mapping them to canonicalized entities in a knowledge base (NED). NER is typically based on trained CRF’s using lexico-syntactic linguistic features. The most popular tool is the Stanford NER Tagger [81]. The best NED methods combine statistical priors about surface names, the contextual similarity between a mention in an input text and descriptions and properties of candidate entities, and the semantic coherence between candidate entities for different mentions. [113–115] are overviews of different methods and tools, and their experimental behavior. The special case of toponym resolution, for geo-entities, exploits spatial relations between candidate places (e.g., their distance). State-of-the-art techniques include [161, 162]. However, the special case of temponym resolution has not received any attention so far.

#### **Time-sensitive Information Retrieval.**

This direction has recently gained much attention, as a substantial fraction of web queries have temporal aspects [95]. Ranking models that capture the temporal scope of queries and documents have been developed in [170–173]. In addition, there is growing interest in the role of time for search-result snippet generation [174], query classification [175, 176], timeline visualization [109, 177, 178], mining web archives and online communities [159, 179], and further tasks in web contents analytics.

## **4.4 SYSTEM OVERVIEW**

Our system takes different text sources (news, biographies, encyclopedic articles) as input. The entire process of temponym resolution is divided into two steps. First, a set of significant phrases are extracted from the input text, performing the *temponym detection*. Second, these phrases are disambiguated onto canonicalized events or facts in the KB, performing the *temponym disambiguation*.



Figure 4.2 illustrates the architecture of our system. The pipeline starts with processing input text documents to detect noun phrases and named entity mentions. By using a mention-entity dictionary [84], we obtain a set of candidate entities for each mention. Similarly, by using a pattern dictionary, we obtain the noun phrases that are possible temponym phrases. By using a knowledge base, we generate candidate mappings from temponym to fact/event. These two sets (mention-to-entity mappings, and temponym-to-fact/event mappings) are later fed into integer linear programs (ILP's). The ILP's use different constraints and objective functions to jointly disambiguate mentions to entities and resolve the temponyms. Finally, the temponyms that are mapped to facts are added to the KB for knowledge enrichment, which is our ultimate goal of populating a knowledge base with this kind of temporal knowledge. This is the flow for the local and joint ILP models we devised for temponym disambiguation. Additionally, in the global model, we also consider a news corpus to mine more relevant cues to enrich the context of temponym candidates and enhance the effectiveness of temponym resolution using ILPs. We explain each part of the pipeline next.

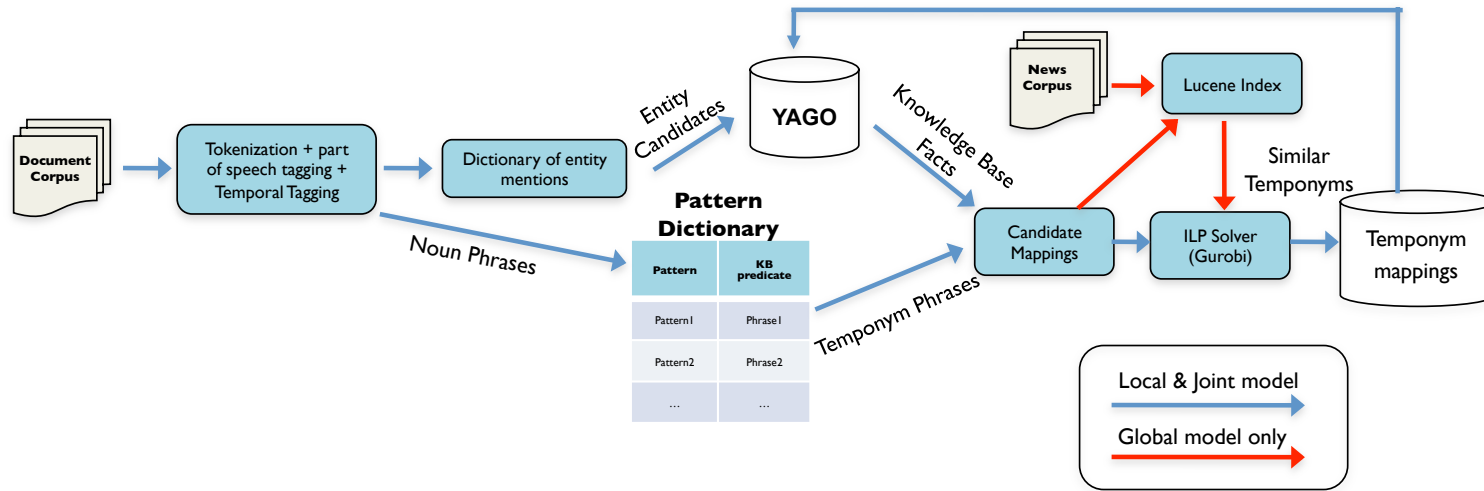


FIGURE 4.2: The processing pipeline for temponym detection and disambiguation.

## 1. Inputs.

The following inputs are used by our system.

- Text inputs: Our system takes arbitrary text documents as input. We use the Stanford NLP software<sup>6</sup> for tokenization and part of speech tagging in input documents.
- Knowledge base: We use the YAGO2 knowledge base to provide us with entities, facts about entities, semantic types of entities. In addition, YAGO2 provides us the mention-entity dictionary which contains textual surface forms (alias names) of entities.
- Repository of relational patterns: Based on the PATTY tool [180], we created a dictionary of lexico-syntactic patterns with semantic type signatures for each of the KB relations. We specifically tailor this repository for events and temporal facts. The details of how this repository is constructed are given in Section 4.5.1.

## 2. Significant Phrase Detection.

We detect several types of textual expressions, each used for later stages:

- TempEx's: We identify TempEx's by using the HeidelTime tagger [98]. The TempEx's are later used to construct a time histogram for the input document, which are explicit cues for temponym resolution.
- Mentions: We detect entity mentions by using the Stanford NER tool [81]. These are later mapped to canonical entities in the knowledge base during joint disambiguation of temponyms and entities. A bottleneck of the NER tool is that it recognizes only mentions that are properly capitalized. However, – in contrast to persons, organizations, locations – the event entities usually are not properly capitalized in text. For instance, the phrase “*The United States presidential election of 2016*” is not detected by the NER tool, although it is an event entity. Our temponym detection stage compensates for this limitation. It extracts such mentions for the entities of type event with a good precision and recall.
- Temponyms: These are the most significant phrase that we aim to detect in text. We explain the temponym detection in details in Section 4.5.

---

<sup>6</sup>[nlp.stanford.edu/software/corenlp.shtml](http://nlp.stanford.edu/software/corenlp.shtml)

The following example shows a sentence and the significant phrases detected in it.

■ **Example 4.4.1** “Ronaldo’s performance during the Euro 2016 tournament made him the best forward of the tournament”

- TempEx’s: “2016”.
- Mentions: “Ronaldo”, “Euro”
- Temponyms: “the Euro 2016 tournament”, “the tournament”

Note that significant phrases might overlap; the disambiguation stage of our system chooses a consistent subset of phrases mapped to semantic targets in the KB.

### 3. Candidate generation.

In this step, our system generates two kinds of candidates.

- entity candidates for mentions.
- event or fact candidates (along with their time scopes) for temponyms.

The entity candidate generation step is guided by the entity-mention dictionary. For example, consider in the example above the candidate entities for the mention “Ronaldo” are ⟨Ronaldo⟩, ⟨Cristiano Ronaldo⟩, ⟨Ronaldo Guiaro⟩, ⟨Ronaldo Rodrigues de Jesus⟩, ⟨Ronaldo Soares Giovanelli⟩, ⟨Ronaldo Morais Silva⟩, and 44 more.

The candidate generation of events/facts for temponyms is guided by the repository of relational patterns. These patterns decide if a temponym phrase denotes an event or a fact. If it is an event, the Wordnet class of the event is determined. For example, in the sentence above, for the temponym “the tournament” the possible candidates are all the event entities in the knowledge base that belong the class ⟨tournament⟩. If the pattern repository yields that a temponym phrase denotes a fact, the predicate of the fact is determined. All the facts having this particular predicate and having one of the candidate entities as subject are candidate facts.

### 4. Outputs.

Our system produces two significant outputs:

- Mentions of (non-event-type) entities are disambiguated to canonical entities.

- Temponyms are mapped to the named events or to facts in the KB.

These two tasks are coupled and jointly solved by our ILP methods. All resolved temponyms are added to the KB. Thus, we provide *knowledge enrichment* in three ways: i) disambiguating the mentions and temponyms in the text and creating semantic markup, ii) temporally annotating the temponyms in the text, and iii) extending the KB with new `<owl:sameAs>` triples that have a temponym phrase as subject, and an event or a fact as object.

## 4.5 TEMPONYM DETECTION

Temponyms typically appear as noun phrases in text. A noun phrase can generally refer to the followings:

- i) the name of an entity (e.g., “president Obama” or “the US president”),
- ii) a class (type) of entities (e.g., “US presidents” or “football clubs”),
- iii) a general concept (e.g., “the climate change” or “linear algebra”),
- iv) textual patterns with temporal scope (e.g., “Greek referendum” or “the FIFA final”),
- v) miscellaneous cases (e.g., idioms, quotes, etc.)

In order to gather noun phrases for case (iv) from a given input text, we use a small number of handcrafted regular expressions over word sequences and their part-of-speech (POS) tags (i.e., word categories). For example, the regular expression `[DT]* [JJ]* [NN]+` matches all phrases optionally start with an article (POS tag DT), optionally have one or more adjectives (POS tag JJ), and are followed by one or more nouns (POS tag NN). Examples are “the next presidential election” or “a most memorable champions league final”.

These regular expressions are fairly liberal so as not to miss any candidate phrases. Only a small portion of the captured noun phrases are indeed temponyms. The temponyms are differentiated from the other cases by harnessing lexico-syntactic patterns for binary relations, as discussed in Section 4.5.1 below. These patterns are chosen conservatively to eliminate false positives.

### 4.5.1 Lexico-Syntactic Patterns

To detect temponyms, we leverage known temporal facts and events in the knowledge base that have associated temporal scopes.

- *Temporal facts*: these are the facts about entities that the relationship between entities holds for a certain time period. Such relationships are represented through temporal predicates e.g.,  $\langle \text{spouse} \rangle$ ,  $\langle \text{hasAcademicAdvisor} \rangle$ ,  $\langle \text{isCeoOf} \rangle$ ,  $\langle \text{playsFor} \rangle$ , etc.
- *Events*: these are the named event entities in the knowledge base that are associated with semantic event types e.g.,  $\langle \text{election} \rangle$ ,  $\langle \text{cup\_final} \rangle$ ,  $\langle \text{protest} \rangle$ ,  $\langle \text{music\_album\_release} \rangle$ , etc.

We create a dictionary of lexico-syntactic patterns with type signatures, and these patterns serve as cues detecting the above two cases and distinguishing them.

#### 4.5.1.1 Patterns for Events

We start processing the large pool of noun phrases detected in input texts. In order to detect the event temponyms, we run all the noun phrases through a noun group parser that determines their semantic head words (e.g., “referendum” in “Greek bailout referendum” or “victory” in “Germany’s victory in Maracana”). The head word of a noun tells us a lot about its semantic type. Therefore, we use these head words to test whether a noun phrase falls into a class of event types. If it does, then it is a candidate temponym denoting a named event. We considered mapping the head words of noun phrases onto WordNet [46]. However, we observed that WordNet has some misleading information in its type hierarchy, for example, placing movies under the type of events. Obviously, a phrase like “the movie about Ronaldo’s life” which has “movie” as the head word is not desired to be a temponym.

To overcome this problem, we harness the knowledge base instead, specifically YAGO2. There are three relations in YAGO2 that only accept an event as their domain and a date as their range. These predicates are  $\langle \text{startedOn} \rangle$ ,  $\langle \text{endedOn} \rangle$ ,  $\langle \text{happenedOn} \rangle$ . Since Yago has strong type checking constraints, the instances of these predicates are very precise, above 95% [18]. We use the names of the left-hand arguments of the facts in these three predicates to generate a list of noun-phrase head words that denote events with high probability, for example, “*election, festival, final*” etc. However,

this approach still yields false positives. For example, we derive “*woodstock*” from “Woodstock” which is a festival, and “*concert*” from “Concert of Europe” which is a conference.

We cope with such false positives as follows: We observe that noun-phrase head words that denote events mostly agree with the YAGO2 event categories. The YAGO2 event categories in turn yield additional cues for deciding whether a noun phrase is an event or not. Examples are “Elections in 1981, Electronic music festivals in Turkey, political scandals” etc. The above case of “Woodstock” is in categories like “rock festivals”, “music festivals” etc., and these are the cues to pick up only “*festival*” as an event type.

This observation is leveraged as follows:

- The YAGO2 event category names are parsed, and the names of their instances (compiled from the predicates  $\langle \text{startedOn} \rangle$ ,  $\langle \text{endedOn} \rangle$ ,  $\langle \text{happenedOn} \rangle$ ) are parsed as well, giving us head words for both categories and instances.
- We compute a mutual information (MI) measure between category-level and instance-level head words, based on:

$$P[\text{eventHead}|\text{catHead}] = \frac{P[\text{eventHead}, \text{catHead}]}{P[\text{catHead}]}$$

The MI scores are later used during temponym resolution as semantic similarity measure.

- We estimate the instance-level head words that denote events with mutual information above a threshold. This way, many false positives coming from specific event names are removed by incorporating the information from event category names.

Using these techniques, we generate a list of noun-phrase head words that denote events with very high probability. This method resulted in a dictionary for type predicates that maps nouns to event types. The dictionary has 370 patterns in total. Examples of event and fact patterns are shown in Table 4.3.

#### 4.5.1.2 Patterns for Temporal Facts

We harness the PATTY repository [180] for the subset of predicates that have temporal scopes, focusing on noun phrases. PATTY contains a total of 160 patterns for temporal

predicates. We keep only patterns above a certain confidence as computed by PATTY. These confidence scores are later used during temponym resolution as semantic similarity measure. A PATTY pattern is a tuple that maps a textual pattern to a KB predicate. An example is:  $\langle \text{“}[[prp]] \text{ marriage with”}, \langle \text{marriedTo} \rangle \rangle$

In our case, we are only interested in the head compound of a PATTY pattern. So, in our setting, this pattern becomes like the following:

$\langle \text{“marriage”}, \langle \text{marriedTo} \rangle \rangle$ . Table 4.3 shows more examples of such patterns together with domain and range specifications.

Nominal pattern	KB Predicate	Domain	Range
“receiving”	$\langle \text{hasWon} \rangle$	$\langle \text{person} \rangle$	$\langle \text{award} \rangle$
“nomination”	$\langle \text{hasWon} \rangle$	$\langle \text{person} \rangle$	$\langle \text{award} \rangle$
“inauguration”	$\langle \text{holdsPosition} \rangle$	$\langle \text{person} \rangle$	$\langle \text{politicalPost} \rangle$
“presidency”	$\langle \text{holdsPosition} \rangle$	$\langle \text{person} \rangle$	$\langle \text{politicalPost} \rangle$
“presidency”	$\langle \text{isLeaderOf} \rangle$	$\langle \text{person} \rangle$	$\langle \text{yagoLegalActorGeo} \rangle$
“death”	$\langle \text{diedOn} \rangle$	$\langle \text{person} \rangle$	$\langle \text{xsd:date} \rangle$
“death”	$\langle \text{diedIn} \rangle$	$\langle \text{person} \rangle$	$\langle \text{city} \rangle$
“death”	$\langle \text{rdf:type} \rangle$	$\langle \text{rdfs:resource} \rangle$	$\langle \text{death} \rangle$
“inauguration”	$\langle \text{rdf:type} \rangle$	$\langle \text{rdfs:resource} \rangle$	$\langle \text{inauguration} \rangle$

TABLE 4.3: Examples of lexico-syntactic patterns for temponym detection.

### 4.5.2 Temponym Features

Our system applies lexico-syntactic patterns on the extracted noun phrases from the input text, and extracts the temponym phrases with the following features:

1. the KB predicates matching the temponym based on the paraphrase dictionary, along with confidence scores;
2. the TempEx’s appearing in the context of the temponym;
3. the entity mentions in the context of the temponym;
4. the sentence that the temponym appears in;
5. the provenance info about the temponym: document URL, (publication) times-tamp of document, etc.



## 4.6 CANDIDATE MAPPINGS GENERATION

Candidate temponyms detected in text come from with various features. In particular, the entity mentions and predicates are significant. The candidate space is generated with two semantic items: i) candidate entities for mentions, and ii) candidate facts/events for temponyms. These candidates are all retrieved from the KB. **Candidate entities** for mentions are taken from the mention-entity dictionary. For each mention, all the entities are retrieved. **Candidate facts** are retrieved from the KB facts based on the predicates of temponyms obtained from the repository of patterns. **Candidate events** are special case of candidate facts that have the predicate as  $\langle \text{rdf:type} \rangle$  and the range as one of the Wordnet event classes like  $\langle \text{election} \rangle$ ,  $\langle \text{tournament} \rangle$ , or  $\langle \text{festival} \rangle$ . Candidate events are retrieved from the KB based on the range signature of the temponym.

- R A temponym phrase might have several predicates in its feature set and these predicates might be for facts or for events. The disambiguation stage of temponym resolution takes care of this and maps at most one predicate for each temponym.

The candidate space can be very large as it is shown in the following example. The example shows the large number of candidate entities, facts, and events.

- Example 4.6.1 “Real Madrid and Portugal forward Ronaldo is the recipient of the Golden Ball for a second consecutive time due to his performance in last season’s Champions League tournament where he set a record by scoring 17 goals.”

mention	# of candidate entities
“Real Madrid”	242
“Portugal”	3 060
“Ronaldo”	130
“Golden Ball”	26
“Champions League”	1227
temponym	# of candidate facts/events
“the recipient of the Golden Ball”	120 529
“Champions League tournament”	4 182

### **Pruning Strategies.**

One of the challenges of resolving temponyms is the large number of candidate entities per mention and candidate facts/events per temponym. To address this problem we prune the candidates that have low potential of being the ideal mapping. In this regard, we refer to the named entity dictionary from [84] to obtain a score for the candidate entities. Using this scoring we rank and select top-k candidates. Consequently, we derive the candidate facts and events from YAGO2 based on these entities. A fact from YAGO2 is a candidate fact if it contains an entity from the above mentioned top-k selected candidates as a subject or an object. In addition, a YAGO2 event is considered a candidate mapping if the subject is one of the top-k entities. This gives us a much smaller set of candidate mappings for a temponym.

The final goal is to select the best mapping among the chosen candidates. For this purpose, we derive several measures of relatedness based for the mappings on diverse features such as textual, temporal and semantic similarity as explained in details next.

## **4.7 TEMPONYM DISAMBIGUATION**

The final task in temponym resolution is to disambiguate them onto KB facts and events. We define the *temponym disambiguation* task as follows:

**Definition 4.7.1 — temponym disambiguation.** Given a set of temponyms together with their contextual cues (entity mentions, TempEx’s, etc.), temponym disambiguation is to map them onto the KB facts and events (while disambiguating the mentions in the context onto KB entities).

Temponym disambiguation is the most difficult task of temponym resolution. This difficulty is reflected in the limitations of using existing NERD techniques to resolve temponyms. We make a vital observation that NERD techniques typically fail to capture named events. This is due to the fact that unlike people, organizations and places the temponyms are not always capitalized. For example, the expressions like “the first Winter Olympics hosted by Russia”, are cannot be resolved by NERD techniques. As we experimentally show in Section 4.9, it is evident that, NED tools such as AIDA miss significant fraction of temponyms. Moreover, the previous NERD techniques cannot cope with temponyms that are facts such as “Obama’s presidency” at all. To this end, we posit that considering entity coherence and temporal coherence jointly is crucial in resolving temponyms that are otherwise ignored. Therefore, when resolving a temponym, our methods also jointly resolve the following:

1. the *entity mentions* in the context of the temponym are disambiguated;
2. the *time point or period* attached to the KB fact or event is propagated to the temponym as additional temporal markup for the input text;
3. the *semantic type* of the temponym is determined (e.g., marriage, election, concert, etc.).

In the rest of this Section, we define several measures that capture the similarities between temponym candidates and KB facts, coherence of entities, temporal expressions and events. Then, we formulate the problem of resolving temponyms as three different Integer Linear Programs (ILP) with an objective to maximize the similarity and coherence. We also introduce several constraints to ensure that the selected mapping is meaningful.

### 4.7.1 Similarity Metrics and Relatedness Measures

We leverage the feature sets of temponyms to derive similarity metrics and relatedness measures. These are:

1. fact-temponym relatedness,
2. mention-entity prior, and
3. coherence measures.

#### 4.7.1.1 Fact-temponym Relatedness

**Definition 4.7.2 — fact-temponym relatedness.**  $w-rel_{tf}$  measures how related a temponym  $t$  to a fact  $f$  in terms of the textual, temporal, and semantic similarity between them. Formally,

$$w-rel_{tf} = w-text_{tf} + w-sem_{tf} + w-temp_{tf}$$

Here,  $w-text_{tf}$  is the jaccard string similarity between the tokens of  $t$  and  $f$ .  $w-sem_{tf}$  is the semantic similarity score for the head noun of  $t$  and the predicate of  $f$ . The semantic similarity score is obtained from the pattern dictionary (as explained in Section 4.5.1).  $w-temp_{tf}$  is the temporal similarity of  $f$  and the normalized dates in the context of  $t$ . The temporal similarity between a temponym and a fact is estimated

from the divergence between the distribution of the normalized dates in the context of the temponym, and the time scope of the fact. The time scope of a fact is converted to a uniform distribution of year dates between the beginning and the end of the time scope. We implement these distributions in the form of histograms.

**Definition 4.7.3 — temporal similarity.** The temporal similarity between temponym  $t$  and fact  $f$  is defined as:

$$w-temp_{tf} = 1 - JSD(H_t || H_f)$$

Where,  $JSD$  is the Jensen-Shannon Divergence (i.e., the symmetric extension of the Kullback-Leibler Divergence, hence, a metric), and  $H_t$  and  $H_f$  are the distributions of concrete dates in the contexts of  $t$  and in the time scope of  $f$ . The Jensen-Shannon Divergence is a popular method of measuring the divergence between two distributions and is calculated as

$$JSD(P||Q) = \frac{1}{2}KL(P||M) + \frac{1}{2}KL(Q||M)$$

Where,  $M = \frac{1}{2}(P + Q)$  and  $KL$  is the Kullback-Leibler divergence calculated as  $KL(P||Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}$ .

#### 4.7.1.2 Mention-entity Prior

A temponym can be mapped to a fact/event if the fact/event contains one of the candidate entities as subject or object. Therefore, the relatedness of a mention to an entity plays a significant role. We call the relatedness of a mention to an entity *mention-entity prior* and define it as:

**Definition 4.7.4 — mention-entity prior.**  $w-ned_{me}$  is a probabilistic prior for mapping a mention  $m$  to an entity  $e$ .

Mention-entity prior is computed based on the frequency of a particular mention  $m$  appearing in the inlink anchor texts referring to specific entity  $e$  in Wikipedia [84].

### 4.7.1.3 Coherence Measures

The features defined above are derived from a given single temponym and fact pair locally. We make an important observation that a coherent text from a document contains entities, explicit TempEx’s and temponyms that have high mutual relatedness in terms of their semantic and temporal properties. To exploit this, we introduce measures for semantic coherence between entities and temporal coherence between facts.

**Definition 4.7.5 — entity-entity coherence.**  $w\text{-coh}_{ee'}$  is the precomputed Jaccard coefficient of two entities  $e$  and  $e'$ :

$$w\text{-coh}_{ee'} = \frac{|\text{inlinks}(e) \cap \text{inlinks}(e')|}{|\text{inlinks}(e) \cup \text{inlinks}(e')|}$$

where *inlinks* are the incoming links in the Wikipedia articles for the respective entity.

The semantic coherence enhances the coherent mapping of mentions to semantically related entities. For example, in the example text in Figure 4.1, the semantic coherence encourages to disambiguate the phrase “Ronaldo” as Portuguese footballer Cristiano Ronaldo rather than the famous Brazilian footballer Ronaldo who has a similar career as Cristiano Ronaldo.

**Definition 4.7.6 — temporal coherence.**  $w\text{-temp}_{ff'}$  is the Jensen-Shannon Divergence between the histograms for the temporal scopes of two facts  $f$  and  $f'$ :

$$w\text{-temp}_{ff'} = 1 - \text{JSD}(H_f || H_{f'})$$

The temporal coherence enhances mapping of temponyms to facts that their time scopes are temporally coherent. For example, in the example text in Figure 4.1, the temporal coherence encourages to disambiguate the phrase “World Cup” as FIFA World Cup 2014 rather than FIFA World Cup 2010, since the latter is temporally incoherent with other facts.

Figure 4.3 depicts the similarity and coherence measures for the temponym “recipient of the Golden Ball” along with the mentions and TempEx’s in its context. All the edges in the figure are weighted, however, they are not shown for the clarity of the image.

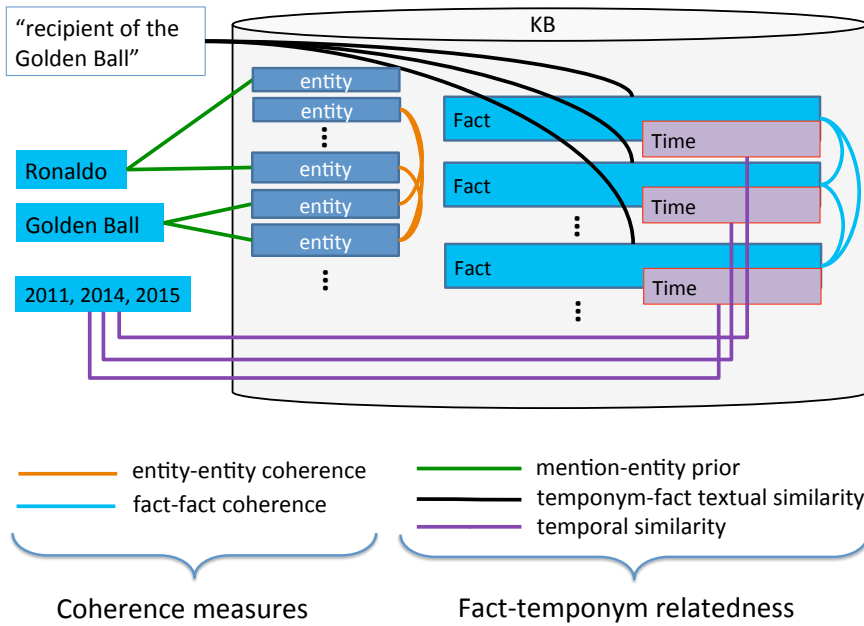


FIGURE 4.3: Similarity and coherence measures.

## 4.7.2 Joint Disambiguation

Having defined several measures that give weights to the temponym fact pairs, the final step is to choose the pair that optimally match. For this purpose, we developed three Integer Linear Program (ILP) models that differ in their scopes for mapping temponyms to the KB.

### 4.7.2.1 Local Model

The local model considers a single temponym as input, and uses the mention-entity prior and the temponym-fact relatedness measure to score candidate mappings. We define the ILP variables, the objective function, and the constraints as follows.

The local model jointly disambiguates entities and the mappings of temponyms by maximizing the total sum of the fact-temponym relatedness and mention-entity prior. It enforces hard constraints to ensure a consistent set of mappings and disambiguations. Constraint 1 ensures that a temponym is mapped to at most one fact. Thus, the representation of the temponym is canonicalized. Constraint 2 ensures that a mention is disambiguated to at most one entity. Finally, Constraint 3 ensures that if a temponym

**Variables**

$X_{tf}$  : 1 if temponym  $t$  is mapped to fact  $f$ , 0 else.

$Y_{me}$  : 1 if mention  $m$  is disambiguated as entity  $e$ , 0 else.

**Objective**

maximize

$$\sum_{t \in T} X_{tf} \times w\text{-rel}_{tf} + \sum_{m \in M} Y_{me} \times w\text{-ned}_{me}$$

**Constraints**

1.  $\sum_f X_{tf} \leq 1, \forall t$

2.  $\sum_e Y_{me} \leq 1, \forall m$

3.  $X_{tf} \leq \sum_{m, e \in \text{args}(f)} Y_{me}, \forall t, f$

---

TABLE 4.4: ILP for the local model.

is mapped to a fact, then the fact should contain a disambiguated entity as its subject or object. This couples the two disambiguation tasks: entity disambiguation and temponym disambiguation. Indeed, a better disambiguation of mentions yields a better mappings of temponyms, since a temponym can be mapped to only a fact of a disambiguated entity.

**4.7.2.2 Joint Model**

Coherence is a significant property because most texts talk about semantically and related stories. Moreover, these stories are temporally coherence. Therefore, the entities and temponyms appearing in a document are supposed to be temporally and semantically coherent. This is the intuition behind the joint model. The joint model extends the local model to jointly resolve all temponyms and disambiguate entities in a given document together by considering semantic coherence between entities and temporal coherence between facts.

The objective, variables and constraints from the local model are borrowed for the joint model. In addition, new objectives to maximize entity coherence and temporal coherence, corresponding variables and constraints are introduced as in Table 4.5:

The joint model disambiguates entities and selects the mappings of temponyms by maximizing the total sum of the fact-temponym relatedness, mention-entity prior, entity-entity coherence, and temporal coherence of facts that are chosen. It enforces hard constraints to ensure a consistent set of mappings. Constraints 1, 2, 3 are the same

---

**Variables**

$X_{tf}$  : 1 if temponym  $t$  is mapped to fact  $f$ , 0 else.

$Y_{me}$  : 1 if mention  $m$  is disambiguated as entity  $e$ , 0 else.

$Z_{ee'}$  : 1 if both disambiguations  $m \rightarrow e$  and  $m' \rightarrow e'$  are selected, 0 else.

$C_{ff'}$  : 1 if both mappings  $t \rightarrow f$  and  $t' \rightarrow f'$  are selected, 0 else.

**Objective**

maximize

$$\sum_{t \in T, f} X_{tf} \times w\text{-rel}_{tf} + \sum_{m \in M, e \in E} Y_{me} \times w\text{-ned}_{me} + \sum_{e, e'} Z_{ee'} \times w\text{-coh}_{ee'} + \sum_{f, f'} C_{ff'} \times w\text{-temp}_{ff'}$$

**Constraints**

1.  $\sum_f X_{tf} \leq 1, \forall t$
  2.  $\sum_e Y_{me} \leq 1, \forall m$
  3.  $X_{tf} \leq \sum_{m, e \in \text{args}(f)} Y_{me}, \forall t, f$
  4.  $Z_{ee'} \leq Y_{me}, \forall m, m', e'$
  5.  $Z_{ee'} \leq Y_{m'e'}, \forall m, m', e'$
  6.  $Z_{ee'} + 1 \geq Y_{me} + Y_{m'e'} \forall m, m', e'$
  7.  $C_{ff'} \leq X_{tf}, \forall t, f, t', f'$
  8.  $C_{ff'} \leq X_{t'f'}, \forall t, f, t', f'$
  9.  $C_{ff'} + 1 \geq X_{tf} + X_{t'f'}, \forall t, f, t', f'$
- 

TABLE 4.5: ILP for the joint model.

as for the local model. Constraints 4, 5, 6 ensure that for any selected pair of entities, the respective mention-entity disambiguations should be selected, too. Constraints 7, 8, 9 ensure that for any selected pair of facts, their respective temponym-fact mappings should be selected, too.

### 4.7.2.3 Global Model

Temponyms often occur (possibly in different forms) in multiple documents of a corpus, e.g., news articles in a news corpus. Therefore, we want to leverage this richer context across documents. The goal of the global model is to process all relevant temponyms and surrounding entity mentions from different documents. The cues obtained this way enrich the context of temponyms from cues across different documents. For example, we can combine cues about several temponyms such as “*German triumph in Maracana*”, “*2014 FIFA World Cup Final*” and “*Argentina vs Germany (2014 FIFA World Cup)*” that refer to same event, from different documents, to enrich the context for resolving each of these temponyms.



However, finding relevant temponyms is expensive if every pair of temponyms are considered. Moreover, it is computationally expensive to feed all potential cues to the ILP solver. Therefore, we need to restrict the search space to highly similar temponym candidates. To achieve this, we index all temponyms phrases obtained from a given news corpus<sup>7</sup> with their contextual features.

The global model employs a grouping function that takes a temponym as input and returns a group of temponyms with high similarity. For this grouping we use the following features:

1. the surface strings of temponyms,
2. the entity mentions in their contexts,
3. the normalized TempEx's in their contexts,
4. the sentences containing the temponyms.

These features one-to-one correspond to the features we extracted for each detected temponym as explained in Section 4.5.2.

This task is realized by indexing all the features mentioned above using the Lucene search engine<sup>8</sup>. For each temponym  $t$  of interest, we run a multi-field boolean search over the different features of the temponym, retrieving a set  $S_t$  of similar temponyms:

$$S_t = \{t' : sim_{Lucene}(t, t') \geq \tau\}$$

where  $sim_{Lucene}$  is the similarity score of the boolean vector space model provided by Lucene and  $\tau$  is a specified threshold. Specifically, the similarity score is computed as:

$$sim_{Lucene}(t, t') = \sum_i \frac{v(t_i) \cdot v(t'_i)}{|v(t_i)| |v(t'_i)|}$$

where  $t_i$  is the vector for feature group  $i$  (string, mentions, TempEx's, sentence) of temponym  $t$ .

For each temponym  $t$  the context features from the temponyms in  $S_t$  are merged. Thus, each temponym is enriched with the contextual information taken from highly similar

<sup>7</sup>We used GDELT (<http://gdeltproject.org/>) news dataset.

<sup>8</sup><https://lucene.apache.org/>

temponyms in the corpus. Then, the same ILP for the joint model is used to compute a solution for the global model. The data flow for the global model is illustrated in Figure 4.2.

## 4.8 POPULATING THE KB WITH TEMPONYS.

The particular KB we populate with canonicalized temponyms is YAGO2. In order to store temponyms in YAGO2, we need to define a schema. A temponym is defined as an entity of class `<yagoTemponym>` that is a sub-class of `<owl:Thing>`. We used existing YAGO2 predicates (relations) if applicable to our setting. Examples are `<startedOn>`, `<endedOn>`, `<rdfs:label>`, and `<rdf:type>`. Then, we define additional predicates with their domain and range specifications as shown in Table 4.6. The predicates `<validDuring>`, `<validBefore>`, and `<validAfter>` are particularly used for the *knowledge linking* task that will be introduced in Section 4.9.2.

Predicate	Domain	Range
<code>&lt;hasID&gt;</code>	<code>&lt;yagoTemponym&gt;</code>	<code>&lt;xsd:integer&gt;</code>
<code>&lt;extractedFrom&gt;</code>	<code>&lt;yagoTemponym&gt;</code>	<code>&lt;xsd:string&gt;</code>
<code>&lt;hasContextEntity&gt;</code>	<code>&lt;yagoTemponym&gt;</code>	<code>&lt;owl:Thing&gt;</code>
<code>&lt;hasContextDate&gt;</code>	<code>&lt;yagoTemponym&gt;</code>	<code>&lt;xsd:string&gt;</code>
<code>&lt;mappedToFact&gt;</code>	<code>&lt;yagoTemponym&gt;</code>	<code>&lt;rdf:Statement&gt;</code>
<code>&lt;validDuring&gt;</code>	<code>&lt;rdf:Statement&gt;</code>	<code>&lt;rdf:Statement&gt;</code>
<code>&lt;validBefore&gt;</code>	<code>&lt;rdf:Statement&gt;</code>	<code>&lt;rdf:Statement&gt;</code>
<code>&lt;validAfter&gt;</code>	<code>&lt;rdf:Statement&gt;</code>	<code>&lt;rdf:Statement&gt;</code>

TABLE 4.6: The set of predicates to store temponyms in YAGO2.

## 4.9 EVALUATION

In order to extensively evaluate our methods, we composed four hypotheses:

1. **Detection quality:** Our methods detect temponyms that are either events or facts with a significant coverage.
2. **Disambiguation quality:** Our methods significantly resolve the temponyms for different kinds of text.

3. **Temporal enrichment:** Our methods substantially add temporal information to documents by finding the temporal scopes of temponyms.
4. **Knowledge enrichment:** Our methods substantially add new knowledge to a knowledge base i) by finding new alias names for events and facts, and ii) by finding the time scope of knowledge base facts via anchoring them to resolved temponyms.

Since each hypothesis aims a different research goal, we developed different experimental settings to effectively assess each hypothesis independently. We first introduce the different datasets we used in the experiments.

### 4.9.1 Datasets

To evaluate the quality of our methods for temponym resolution, we performed experiments with three datasets with different characteristics: WikiWars, Biographies, and News.

#### **WikiWars.**

The WikiWars corpus [181] has been popular in benchmarks for temporal tagging (i.e., resolving explicit, relative and implicit TempEx's). It contains 22 long Wikipedia articles about major wars in human history. These articles are specifically rich in terms of TempEx's and named events. Thus, the temponyms detected in these articles are mostly of the event type. Note that WikiWars articles are plain text documents that do not contain any structured elements of Wikipedia such as entity links, categories, etc.

#### **WikiBios.**

These are Wikipedia articles on the biographies of 30 prominent politicians (e.g., Barack Obama, Hugo Chávez, Vladimir Putin). We refer to this dataset as *WikiBios*. In contrast to the WikiWars, this corpus contains fewer event temponyms but features many temponyms that refer to temporal facts (awards, spouses, positions held, etc.). This makes it particularly challenging, since spotting facts is harder than spotting events which is a specific case of named entity disambiguation task. Note that WikiBios articles are plain text documents that do not contain any structured elements of Wikipedia such as entity links, categories, etc.

#### **News articles.**

We show that our methods can perform well not only on properly edited text that are rich in terms of events and facts (i.e. WikiWars, WikiBios) but also on the

news that are compiled from a large source of news channels. We used GDELТ (<http://gdelтproject.org/>) news dataset for our experiments. GDELТ contains a set of entities for each article; however, we ignored these annotations and solely relied on our own methods to extract and disambiguate entities. In total, this test corpus contains 1,5 million news articles.

## 4.9.2 Evaluation Tasks and Metrics

To validate each hypothesis explained above we define an evaluation task.

### 4.9.2.1 Temponym Detection Quality

We evaluated the quality of temponym detection by checking whether a detected noun phrase is indeed a temponym. We divided this task into two separate tasks: Event detection quality, and fact detection quality. For the event detection task, we manually annotated the named events appearing in WikiWars dataset. In total, we annotated 1 154 events. We compare our method’s coverage to the state-of-the-art entity disambiguation tool AIDA. In order to make a fair comparison in favor of AIDA, we only considered the named events that are linked to particular Wikipedia event articles by Wikipedia editors. Thus, we ended up 646 named event phrases with the respective sentences that they appear in.

For the fact detection task we manually annotated the facts appearing in WikiBios dataset. We only considered the first three paragraphs of each article during annotation. We annotated 593 temporal facts. The previous works [35, 40] consider only subject-verb-object style phrases for fact extraction. Since temponyms are of the noun phrase nature, we do not compare our method’s coverage to previous work. Thus, we just report the recall values.

### 4.9.2.2 Temponym Disambiguation Quality

The evaluation of mapping of temponyms is a human intelligence task. It has to be checked whether the mapping makes sense. We evaluated the quality of temponym disambiguation by checking whether a temponym is mapped to the correct event or fact in the KB. This implies that the temporal scoping for the temponym is correct, too. We additionally checked whether the mentions in the temponym context are correctly

disambiguated as well. There is no prior ground-truth for these corpora and creating such a dataset is a big amount of human work. Thus, we manually judged the quality of the computed mappings. We randomly selected 100 temponyms per model per dataset. In other words, 200 temponyms from WikiWars mappings, 300 from WikiBios mappings, and 300 from News mappings, a total of 800 temponym mappings. For statistical significance, we calculated Wilson confidence intervals [182].

We ran the local model, the joint model, and the global model on each corpus with the exception of WikiWars. The global model is not applicable here, as it requires multiple documents on the same or overlapping topics. In contrast, the 22 WikiWars articles are fairly disjoint in their contents and are not mentioned in GDELT news corpus much.

The evaluation is done by marking a mapping with three different scores; Correct, Okay, Wrong. Table 4.11 shows some examples of Correct, Okay, and Wrong matches. A mapping is considered “Okay” if it has partially correct match. For example, the temponym *the second term of Merkel* is mapped to the correct fact  $\langle \text{AngelaMerkel}, \text{holdsPosition}, \text{ChancellorOfGermany} \rangle$  but it is marked as “Okay”. The reason is that the second term of Angela Merkel is actually from 2009 to 2013 rather than from 2005 to now.

Precision is calculated in two different ways:

- For *strict* precision, we count the *Okay* mappings as wrong:

$$Precision_{strict} = \frac{\#Correct}{\#Total\ mappings}$$

- For *relaxed* precision, we count the *Okay* mappings as true:

$$Precision_{relaxed} = \frac{\#Correct + \#Okay}{\#Total\ mappings}$$

### 4.9.2.3 Knowledge Enrichment

The temponym resolution task has two important outcomes in terms of knowledge enrichment: *knowledge paraphrasing* and *knowledge linking*.

**1. Knowledge paraphrasing:** Temponym resolution enriches the KB by providing additional paraphrases for known events and facts. For example, our methods can

add the temponym “*the largest naval battle in history*” as an alias for the event  $\langle \text{BattleOfLeyteGulf} \rangle$ . This is represented as  $\langle \text{BattleOfLeyteGulf} \text{ rdfs:label "the largest naval battle in history"} \rangle$ . This task is called *knowledge paraphrasing*.

We assess the knowledge paraphrasing, by comparing outcome of our methods to to YAGO2 knowledge base in terms of paraphrase coverage. Therefore, we randomly chose 100 correctly mapped temponyms and checked how many temponyms are already known to YAGO2, either as an event entity or as a fact. We built a text index over all events and facts in YAGO2 and their alias names. For the randomly chosen 100 temponyms, we queried this index for each temponym and took the top-10 most relevant results for each query. We manually checked all these returned answers, thus considering also approximate matches for a fair comparison in favor of YAGO2.

**2. Knowledge linking:** Temponym resolution also enhances the fact extraction tools for knowledge bases by providing them additional temporal and semantic clues. For example, in the sentence “Ronaldo joined Real Madrid during second term of Florentino Pèrez” a fact extraction tool can extract the fact  $\langle f1:\text{CristianoRonaldo}, \text{playsFor}, \text{RealMadrid} \rangle$  but no time scope attached. Temponym resolution would normalize the phrase *second term of Florentino Pèrez* to time  $[2009, \text{now}]$  by mapping it to the fact  $\langle f2:\text{FlorentinoPèrez}, \text{isPresidentOf}, \text{RealMadrid}, [2009, \text{now}] \rangle$ . Thus, a fact extraction tool can temporally link two facts as a new fact  $\langle f3:f1, \text{validDuring}, f2 \rangle$ . This task is called *knowledge linking*.

For the knowledge linking task, we carried out an extrinsic case study. We modified the PATTY’s binary fact extraction patterns to ternary patterns so that they can take a temponym as an argument. For example, the PATTY pattern  $\langle \text{subject}, \text{verb}, \text{object} \rangle$  is modified to  $\langle \text{subject}, \text{verb}, \text{object}, \text{preposition}, \text{temponym} \rangle$ . Thus, a fact extracted from  $\langle \text{subject}, \text{verb}, \text{object} \rangle$  triple can be linked to the particular temponym through a particular preposition such as “during, before, after”. For this task, we ran PATTY tool on its extraction corpus. We report the number of facts that are linked to temponyms through three prepositions “during, before, after”.

#### 4.9.2.4 Temporal Enrichment

To show our methods can substantially add extra temporal information to documents, we compare our methods to well known HeidelTime tagger [98] by running the both methods on WikiWars and WikiBios datasets. We compare the number of normalized

Dataset	# TempEx's	# temponyms
<i>WikiWars</i>	2 681	2 504
<i>WikiBios</i>	6 822	5 390
<i>News</i>	258 420	225 371

TABLE 4.7: Number of temponyms and TempEx's detected in each dataset.

TempEx's by HeidelTime tagger to the number of normalized temponyms by our methods.

### 4.9.3 Results

#### 4.9.3.1 Detection quality

Our methods detected 233 265 temponyms from three corpora. Specifically, 2 504 temponyms from WikiWars, 5 390 from WikiBios, and 225 371 from the News dataset are extracted. Table 4.7 shows these values compared with the number of TempEx's appearing in these corpora. These numbers already show that, the temponyms are quite abundant in text. Therefore, they should be given significance as much as temporal expressions.

We calculated the recall values specifically for events in WikiWars – event detection quality –, and for facts in WikiBios datasets – fact detection quality –, Table 4.8.

**i) Event detection.** Among the 646 annotated named events in WikiWars dataset, AIDA detected 186 of them, which results in 29% coverage. On the contrary, our methods detected 338 of the events, which resulted in 52% coverage.

**ii) Fact detection.** Among the 593 annotated temporal facts in WikiBios dataset, our method detected 195 of them, which yields a 33% coverage. It might seem a low coverage. However, considering that temporal facts can be phrased in text in many different ways, our results are encouraging.

Figure 4.4 shows the comparison of AIDA and our methods in terms of detection quality. Firstly, we see that AIDA can perform relatively good on the WikiWars dataset, because the dataset contains mostly named events. Thus, AIDA can detect properly capitalized event names such as “World War II” or “Great Recession”. However, it fails on event names that are long or not properly capitalized like “the first Winter

	WikiWars	WikiBios
# Gold annotations	646	593
# AIDA's extractions	186	–
# Our extractions	338	195

TABLE 4.8: Recall values for AIDA and for our method.

olympics hosted by Russia”. On the contrary, our temponym detection methods can detect such event names successfully. Therefore, temponym detection compensates this limitation of NERD tools. Secondly, AIDA performance on the WikiBios dataset is very poor. The reason is that this dataset mostly contains paraphrase of facts such as “Obama’s presidency” or “Merkel’s second term” etc. On the contrary, our methods detect such phrases. As a result, we observe that general aim NED tools such as AIDA are not well suited for temponym detection. Therefore, specialized solutions such as our methods should be pursued. Table 4.9 shows some interesting temponyms detected by our methods. The first temponym in the table is detected by the both systems, whereas, the last one is not detected by any system.

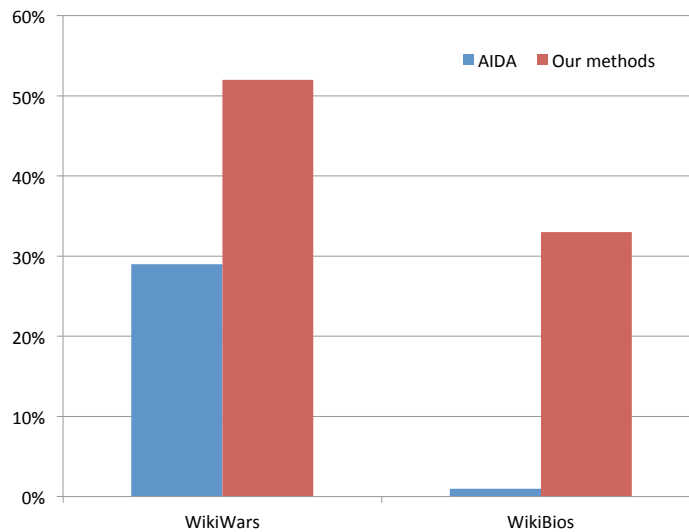


FIGURE 4.4: Example of Text with Temporal Expressions and Mappings to a Knowledge Base.

#### 4.9.3.2 Disambiguation quality

We evaluated the overall disambiguation quality over randomly selected 800 temponym mappings. We computed 95% Wilson confidence intervals for strict precision and



temponym	AIDA	Our method
“Great Recession”	✓	✓
“health care reform”	–	✓
“the first Winter olympics hosted by Russia”	–	✓
“Barack Obama’s graduation”	–	✓
“his childhood”	–	–

TABLE 4.9: Examples of temponyms that are detected or not by our methods and AIDA.

Dataset	Strict			Relaxed		
	Local	Joint	Global	Local	Joint	Global
WikiBios	.54 ±.09	.60 ±.09	.68 ±.09	.61 ±.09	.66 ±.09	.76 ±.08
WikiWars	.75 ±.08	.82 ±.07	n/a	.84 ±.07	.86 ±.06	n/a
News	.58 ±.09	.64 ±.09	.67 ±.09	.69 ±.09	.75 ±.08	.79 ±.08

TABLE 4.10: Precision at 95% Wilson interval for different methods

for relaxed precision, on all three datasets. The strict matching evaluation gives us a  $65\% \pm 0.03$  precision. The relaxed matching evaluation gives us a  $73\% \pm 0.03$ . The detailed precision results for each dataset and for each method are shown in Table 4.10.

We see that the joint and global models boost the precision by a large margin. For the relaxed precision measure, the global models achieved substantial gains over the joint models. The precision numbers are particularly good for the News and the WikiWars corpora, thus achieving high value for semantic markup and knowledge enrichment. For WikiBios, the results are somewhat worse. Here we faced the challenge that many temponyms refer to SPOT facts (e.g., awards, spouses, children, held positions, etc.) rather than typed events, which is much harder to deal with. Nevertheless, the results are very encouraging, given that temponym resolution is more demanding than TempEx resolution and the state-of-the-art results for TempEx’s are 80 to 90% [160]. Examples of temponyms, their mappings by our methods, and the manual evaluation of the mappings are shown in Table 4.11.

temponym	Yago	Our model	Time scope	Eval
<i>“the Great Recession”</i>	⟨GreatRecession⟩	⟨GreatRecession⟩	[2007, 2009]	Correct
<i>“the second term of Merkel”</i>	–	⟨AngelaMerkel, holdsPosition, ChancellorOfGermany⟩	[2005, now]	Okay
<i>“Obama’s graduation”</i>	–	⟨BarackObama, graduatedFrom, HarvardLawSchool⟩	[1991, 1991]	Correct
<i>“the first Winter Olympics to be hosted by Russia”</i>	–	⟨2014WinterOlympics⟩	[2014,2014]	Correct
<i>“Putin’s presidency”</i>	–	⟨VladimirPutin, holdsPosition, PrimeMinisterOfRussia⟩	[2008, 2012]	Wrong
<i>the assassination of Martin Luther King</i>	–	⟨MartinLutherKing, diedOn, 1968-04-04⟩	[1968, 1968]	Correct

TABLE 4.11: Comparison of AIDA vs. our methods for the temponym detection task.

### 4.9.3.3 Temporal enrichment

We compared our best performing model, global model, to HeidelTime tagger to see how much additional temporal information is added to documents. HeidelTime normalized 5 533 TempEx’s from WikiBios dataset, and 2 047 from WikiWars dataset to date values. Whereas, our methods normalized 885 temponyms from WikiBios dataset, and 558 from WikiWars dataset to date values by disambiguating these temponyms to KB facts or events. Note that these temponyms are not detected by HeidelTime tagger at all. Figure 4.5 shows that our methods add 16% additional temporal information to WikiBios dataset and 27% to WikiWars dataset.

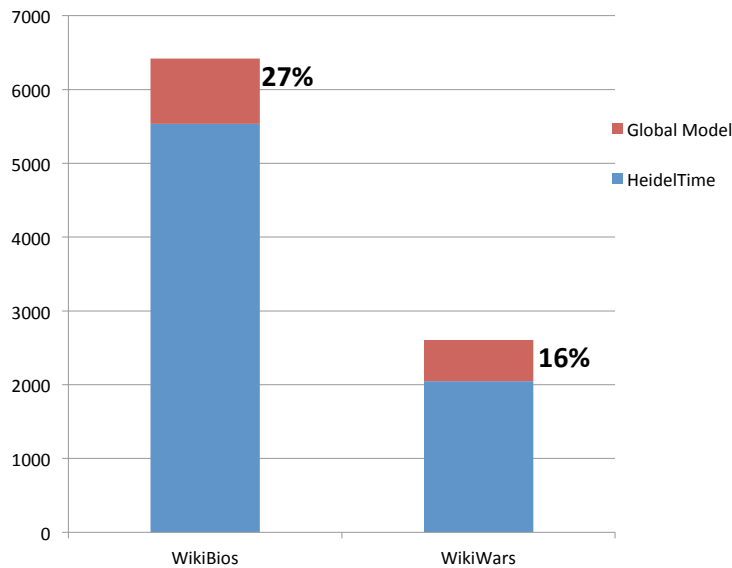


FIGURE 4.5: Comparison of HeidelTime tagger and our methods for the temporal enrichment task.

### 4.9.3.4 Knowledge enrichment

For the knowledge paraphrasing task, the manual assessment over randomly selected 100 temponyms showed that YAGO2 knows only 52 of the events given by the 100 temponyms. On the remaining 48, YAGO2 does not even have any approximate matches. YAGO2’s coverage is great for canonicalized event names such as “the Great Recession, Second World War, etc.” However, it is largely agnostic to phrases for less standardized events such as “the second term of Merkel”, “Obama’s graduation”, “the last presidential election in France”, etc. Our methods do not only detect these temponyms but also disambiguate them correctly onto events or facts. Examples from this comparison are shown in Table 4.11.

For the knowledge linking task, our methods disambiguated 65 625 temponyms surrounding the facts that are extracted by ternary patterns. 12 803 (20%) of these temponyms are temporally linked to the extracted facts through prepositional links.

For the knowledge linking task, our methods disambiguated 65 625 temponyms surrounding the facts that are extracted by ternary patterns. 12 803 (20%) of these temponyms are temporally linked to the extracted facts through prepositional links. For example, the base facts extracted from the sentence “Hillary was First Lady of the United States during Clinton’s tenure.” by this method are

`<f1:HillaryClinton, holdsPosition, FirstLadyOfUS>`,

`<f2:BillClinton, holdsPosition, PresidentOfUS>`.

These two base facts. then, are linked through the reification mechanism of RDFS.

Thus, `<f1>` and `<f2>` are linked as

`<f3:f1, validDuring, f2>`.

#### 4.9.4 Releasing Data

The temponym repository that we created from different corpora is downloadable under the url [www.mpi-inf.mpg.de/yago-naga/evin/](http://www.mpi-inf.mpg.de/yago-naga/evin/). Moreover, the manually annotated datasets for temponym detection is also available under the same url.

## 4.10 APPLICATIONS

State-of-the-art temporal taggers like SUTime [112] and HeidelTime [98] annotate TempEx's with TIME3 tags<sup>9</sup>. However, they cannot handle temponyms. The main reason is that these tools are rule based systems and do not attempt deep analysis of the context of TempEx's. On the contrary, temponyms, in general, can be interpreted correctly given enough contextual cues. For example the temponym “*a second term*” is ambiguous. However, given the sentence “Barack Obama was reelected president and was sworn in for a second term.”, the temponym “*a second term*” has a unique interpretation. It is significant to note that certain kind of temponyms are not much ambiguous. Examples are “*World Cup Final 1998*”, “*Sarkozy's marriage with Carla Bruni*”, etc. These temponyms are explicit enough for a unique interpretation. We call the temponyms that do not require deep contextual analysis for disambiguation *explicit temponyms*. In this section, we extend the publicly available temporal tagger HeidelTime<sup>10</sup> to cover explicit temponyms through particular rules created for explicit temponyms. Thus, the tagger can run stand-alone without distant supervision of a knowledge base and without deep analysis of temponyms.

### 4.10.1 Implementation

#### 4.10.1.1 Explicit Temponym Creation Process

We merge three data sources to create a large repository of explicit temponyms along with their temporal annotations.

1. **YAGO** provides temporal predicates such as  $\langle \text{holdsPosition} \rangle$ ,  $\langle \text{isMarriedTo} \rangle$ , and the facts of such predicates. Moreover, YAGO provides the predicates  $\langle \text{startedOn} \rangle$ ,  $\langle \text{endedOn} \rangle$ ,  $\langle \text{happenedOn} \rangle$  to indicate the start and end dates of named events.
2. **AIDA** contains alias names for entities. For example, “*London Olympic Games*”, “*Games of the XXX Olympiad*” are alias names for the entity  $\langle \text{2012SummerOlympics} \rangle$ .
3. The **pattern dictionary** we created in 4.5.1.

---

<sup>9</sup><http://www.timeml.org>

<sup>10</sup><https://github.com/HeidelTime/heideltime>

### Temponym Repository for Facts.

First, all the temporal facts from YAGO are paraphrased using alias names from the AIDA dictionary. For example, a paraphrase of the fact

`<HillaryRodhamClinton holdsPosition UnitedStatesSecretaryOfState>` is `<Hillary Clinton holdsPosition Secretary of State>` by replacing subject and object entities of the fact with their alias names. The next step is to map the temporal predicates of these facts to the noun phrases in the pattern dictionary. Therefore, the paraphrased fact `<HillaryClinton holdsPosition SecretaryOfState>` is converted to the temponym “*Hillary Clinton’s term as US Secretary of State*” by mapping the predicate `<holdsPosition>` to patterns such as “[*PER*]’s term as”, “[*PER*] serving as”, etc.

### Temponym Repository for Events.

AIDA may contain same alias name for different entities with certain confidence scores attached. In order to avoid the disambiguity, we consider the alias names that are unique to event entities. Thus, we create explicit temponyms of events by taking only canonical alias names of events. For example, the named event `<2012SummerOlympics>` has many alias names. But there are two alias names that are not shared by any other entity in AIDA, namely, “*London Olympic Games*”, and “*Games of the XXX Olympiad*”.

#### 4.10.1.2 Temponym Tagging with HeidelTime

HeidelTime is a rule-based temporal tagger. It contains two important resource files that are responsible for the two stages of temporal expression detection and normalization. *Pattern resources* contain phrases (e.g., “April”) to be detected. *Normalization resources* contain information how to normalize the detected expressions (e.g., “April” to 04). There is one additional resource called *rules* to handle the extraction and normalization jointly. Further details about HeidelTime’s rule syntax is explained in [97, 98].

For temponym tagging, we use explicit temponyms to create HeidelTime pattern and normalization resources. Simple rules combine this information and are applicable in the same way as all rules for standard temporal expressions. Verbal paraphrases are also added. For instance, using the patterns `PER_holdsPosition` (names of persons for the facts of `<holdsPosition>`) and `noun_holdsPosition` (noun phrases such as “[*PER*]’s term as” and “[*PER*] serving as”), the rule `ext='%PER_holdsPosition1`

*%noun\_holdsPosition',norm='normPER\_holdsPosition(1)'* matches phrases about a person's holding a position and assigns respective dates.

#### 4.10.2 Statistics

Overall, we added to Heidelberg's English resources temponyms for more than 40,000 events and for more than 900,000 facts with several paraphrases. Currently, facts about persons (birth, death, political position, marriage; 664,000) and culture (releases, directions, authors; 253,000) are contained. The events cover several types, e.g., sport events and (historic) battles. Note that we only added those temponyms, for which explicit start and end information is available in YAGO.

To test how many temponyms are detected and normalized on an example corpus, we ran Heidelberg on the WikiWars corpus [181]. We extracted a total of 212 temponyms so that about 8% additional temporally annotated phrases are detected in addition to the standard TempEx's.

#### 4.10.3 Software

The Heidelberg tagger extended with explicit temponyms is publicly available under the url [www.mpi-inf.mpg.de/yago-naga/evin/](http://www.mpi-inf.mpg.de/yago-naga/evin/).

## 4.11 SUMMARY

This chapter has addressed the goal of populating knowledge bases with temponyms. In addition, we can enrich text documents by temponyms along with their semantic and temporal annotation. We developed the first model for temponym resolution that uses joint inference for high-quality mappings to a knowledge base.

Our system takes different text sources (news, biographies, encyclopedic articles) as input and proceeds in two steps. First, temponym phrases are detected together with their contextual cues such as entity mentions, and other TempEx's around. Second, these phrases are disambiguated onto canonicalized events or facts in the KB, performing the temponym disambiguation. We developed a set of integer linear programs (ILP's) using different constraints and objective functions to jointly disambiguate mentions to entities and resolve the temponyms.

Our comprehensive experiments with three corpora (biographies, history documents, news articles) demonstrate the viability and quality of our solution. The experiments show that our methods indeed can populate knowledge bases with this new kind of temporal knowledge. Moreover, our methods can substantially enrich text documents with new temporal information that cannot be detected by state-of-the-art tools.

The temponym repository<sup>11</sup> that we created can be used for various applications. As an example, we extended the well known HeidelTime tagger with a subset of temponyms to extract and normalize temponyms as a stand-alone tool.

---

<sup>11</sup>[www.mpi-inf.mpg.de/yago-naga/evin/](http://www.mpi-inf.mpg.de/yago-naga/evin/)





# CHAPTER 5

## CONCLUSION

### 5.1 SUMMARY

This dissertation presented two main contributions to cover a common problem of current knowledge bases, the lack of temporal knowledge. The contributions are in the areas of populating knowledge bases with named events, and with temponyms, the alias names denoting events and temporal facts.

Our first contribution is EVIN for named event extraction from news. EVIN extracts events in the long tail by going beyond Wikipedia and extracting ontological events from news. Thus, it fills an important gap in the *scope* and *freshness* of knowledge bases. EVIN takes news articles from arbitrary sources as input, and distills the extracted cues into canonicalized events along with temporal ordering and semantic labeling. EVIN’s technical contribution here is to capture several similarity measures among news articles in a multi-view attributed graph, and distill named events from this graph by a novel graph coarsening algorithm based on the minimum description length principle.

Our second contribution is populating knowledge bases with temponyms. Thus, we provide knowledge bases with a new kind of temporal knowledge that is essential for semantic search, query understanding, summarization, deep text analytics, KB curation, and other tasks. We present methods for detecting temponyms in text, and mapping them to semantic targets in a knowledge base to canonicalize their representation. Our methods employ integer linear programs for joint entity disambiguation and temponym resolution. This extends the knowledge base by obtaining new alias names for facts and events. In addition, it enriches documents with semantic and temporal annotations.

## 5.2 OUTLOOK

While the methods presented in this dissertation make significant progress toward populating knowledge bases with temporal knowledge, there is still room for future work. We conclude this dissertation by discussing open research directions in temporal knowledge harvesting.

### **Continuous Event Extraction.**

EVIN harvests events in an offline manner from a given news corpus. However, the knowledge about events is very dynamic, as new events emerge continuously. In addition, beyond news, there is *social media* as a rich source for reporting events. The research question here is: how can we extend the methods of EVIN for continuous extraction of events from news and social media? The first challenge here is the continuous update of the multi-view attributed graph. The second challenge is to come up with new coarsening algorithms that can work on a highly dynamic graph. Such a continuous event extraction framework requires maintaining the event groups under dynamic weight updates for directed and undirected edges, for attributes of nodes, and for other elements of the MVAG.

### **Unsupervised Temponym Resolution.**

We defined temponyms as specific kinds of noun phrases that are alias names for events and facts in a knowledge base. Hence, the temporal resolution task is guided by a knowledge base, making it a distantly supervised task. However, knowledge bases are incomplete. As a result, many temporal phrases that are indeed about facts and events are not disambiguated, since knowledge bases may not contain the respective events or facts at all. The research question here is: how can we detect and disambiguate temponyms without relying on a knowledge base, yet with high accuracy? The main problem is to find canonical representations for temponyms without using a knowledge base. The second challenge is to link temponyms to relevant out-of-knowledge-base entities. Resolving temponyms without background knowledge base requires solving these two problems jointly.

# LIST OF FIGURES

3.1	Output for the theme of “ <i>UEFA Champions League 2012/2013</i> ” . . . .	25
3.2	Architecture of the EVIN System. . . . .	28
3.3	Two step of computing semantic types for news. . . . .	32
3.4	Coarsening a multi-view attributed graph. . . . .	41
3.5	Screenshot of EVIN browser. . . . .	67
3.6	Grouping of the news into events. . . . .	68
4.1	Example of Text with Temporal Expressions and Mappings to a Knowledge Base. . . . .	74
4.2	The processing pipeline for temponym detection and disambiguation. . . . .	83
4.3	Similarity and coherence measures. . . . .	95
4.4	Example of Text with Temporal Expressions and Mappings to a Knowledge Base. . . . .	105
4.5	Comparison of HeidelTime tagger and our methods for the temporal enrichment task. . . . .	108



# LIST OF TABLES

2.1	A fragment of the YAGO knowledge base in tabular form. . . . .	9
3.1	Wikipedia infobox and categories for the article French Revolution. . .	26
3.2	Table of the notations used in this work. . . . .	31
3.3	The set of predicates to store events in YAGO. . . . .	51
3.4	Precision, recall, and F1 scores at $k$ , for labeling. . . . .	57
3.5	Precision, recall, and F1 scores for grouping. . . . .	58
3.6	Precision, recall, and F1 scores for chaining. . . . .	58
3.7	Examples of 10 keyword queries used for evaluating events. $P_{Gr}$ stands for the grouping precision, $P_{Ch}$ stands for the chaining precision	61
3.8	Examples of 5 semantic (wordnet) queries used for evaluating events. $P_{Gr}$ stands for the grouping precision, $P_{Ch}$ stands for the chaining precision . . . . .	61
3.9	Statistics for extracted events. . . . .	63
3.10	Top-50 semantic classes populated with events. . . . .	63
3.11	Coverage of events in YAGO vs. EVIN_KB. . . . .	63
3.12	Two sample EVIN_KB events for given queries. YAGO has no results for these queries. . . . .	65
4.1	Excerpt from a biography of Cristiano Ronaldo. . . . .	73
4.2	Examples of SPOT facts. . . . .	80
4.3	Examples of lexico-syntactic patterns for temponym detection. . . . .	89
4.4	ILP for the local model. . . . .	96
4.5	ILP for the joint model. . . . .	97
4.6	The set of predicates to store temponyms in YAGO2. . . . .	99
4.7	Number of temponyms and TempEx's detected in each dataset. . . . .	104
4.8	Recall values for AIDA and for our method. . . . .	105
4.9	Examples of temponyms that are detected or not by our methods and AIDA. . . . .	106
4.10	Precision at 95% Wilson interval for different methods . . . . .	106
4.11	Comparison of AIDA vs. our methods for the temponym detection task.	107



# BIBLIOGRAPHY

- [1] Matthias Steup. Epistemology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition, 2014. (Cited on page [1](#).)
- [2] Simone Paolo Ponzetto and Michael Strube. Deriving a Large Scale Taxonomy From Wikipedia. In *Proceedings of the 22nd National Conference on Artificial Intelligence*, volume 7, pages 1440–1445, 2007. (Cited on page [2](#).)
- [3] Simone Paolo Ponzetto and Michael Strube. Taxonomy Induction Based on a Collaboratively Built Knowledge Repository. *Artificial Intelligence*, 175(9-10): 1737–1756, 2011. (Cited on page [2](#).)
- [4] Fabian M Suchanek, Gjergji Kasneci, and Gerhard Weikum. YAGO: A Core of Semantic Knowledge Unifying WordNet and Wikipedia. In *Proceedings of the 16th International Conference on World Wide Web*, pages 697–706, 2007. (Cited on pages [2](#), [8](#), [11](#), [17](#), [21](#), [26](#), [32](#), and [35](#).)
- [5] Fei Wu and Daniel S Weld. Automatically Refining the Wikipedia Infobox Ontology. In *Proceedings of the 17th International Conference on World Wide Web*, pages 635–644, 2008. (Cited on page [2](#).)
- [6] Eugene Agichtein and Luis Gravano. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the 5th ACM Conference on Digital Libraries*, pages 85–94, 2000. (Cited on pages [2](#) and [17](#).)
- [7] Philip Bohannon, Nilesh Dalvi, Yuval Filmus, Nori Jacoby, Sathiya Keerthi, and Alok Kirpal. Automatic Web-Scale Information Extraction. In *Proceedings of the 2012 ACM SIGMOD International Conference on Management of Data*, pages 609–612, 2012. (Cited on pages [2](#) and [17](#).)
- [8] Sergey Brin. Extracting Patterns and Relations from the World Wide Web. In *International Workshop on the World Wide Web and Databases*, pages 172–183, 1998. (Cited on pages [2](#) and [17](#).)



- [9] Michael J Cafarella. Extracting and Querying a Comprehensive Web Database. In *Proceedings of the 4th Biennial Conference on Innovative Data Systems Research*, 2009. (Cited on pages 2 and 17.)
- [10] Andrew Carlson, Justin Betteridge, Bryan Kisiel, Burr Settles, Estevam R Hruschka Jr, and Tom M Mitchell. Toward an Architecture for Never-Ending Language Learning. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, volume 5, page 3, 2010. (Cited on pages 2, 8, and 17.)
- [11] Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Sriram Raghavan, Frederick R Reiss, and Shivakumar Vaithyanathan. Systemt: An Algebraic Approach to Declarative Information Extraction. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 128–137, 2010. (Cited on pages 2 and 17.)
- [12] Philipp Cimiano and Johanna Völker. Text2Onto. In *Natural Language Processing and Information Systems*, pages 227–238. 2005. (Cited on pages 2 and 17.)
- [13] AnHai Doan, LGR Ramakrishnan, and S Vaithyanathan. Introduction to the Special Issue on Managing Information Extraction. *SIGMOD Record*, 37(4), 2009. (Cited on pages 2 and 17.)
- [14] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Unsupervised Named-Entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91–134, 2005. (Cited on pages 2 and 17.)
- [15] Yuan Fang and Kevin Chen-Chuan Chang. Searching Patterns for Relation Extraction over the Web: Rediscovering the Pattern-Relation Duality. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 825–834, 2011. (Cited on pages 2 and 17.)
- [16] Tim Furche, Georg Gottlob, Giovanni Grasso, Omer Gunes, Xiaoan Guo, Andrey Kravchenko, Giorgio Orsi, Christian Schallhart, Andrew Sellers, and Cheng Wang. DIADEM: Domain-Centric, Intelligent, Automated Data Extraction Methodology. In *Proceedings of the 21st International World Wide Web Conference, Companion Volume*, pages 267–270, 2012. (Cited on pages 2 and 17.)

- [17] Georg Gottlob, Christoph Koch, Robert Baumgartner, Marcus Herzog, and Sergio Flesca. The Lixto Data Extraction Project: Back and Forth between Theory and Practice. In *Proceedings of the 23rd ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, pages 1–12, 2004. (Cited on pages 2 and 17.)
- [18] Johannes Hoffart, Fabian M. Suchanek, Klaus Berberich, and Gerhard Weikum. YAGO2: A Spatially and Temporally Enhanced Knowledge Base from Wikipedia. *Artificial Intelligence*, 194:28–61, 2013. (Cited on pages 2, 8, 12, 13, 17, 21, 26, 32, 50, 62, 77, 79, and 87.)
- [19] Raphael Hoffmann, Congle Zhang, and Daniel S Weld. Learning 5000 Relational Extractors. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 286–295, 2010. (Cited on pages 2 and 17.)
- [20] Sebastian Krause, Hong Li, Hans Uszkoreit, and Feiyu Xu. Large-Scale Learning of Relation-Extraction Rules with Distant Supervision from the Web. In *Proceedings of the 11th International Semantic Web Conference*, pages 263–278, 2012. (Cited on pages 2 and 17.)
- [21] Nicholas Kushmerick. *Wrapper Induction for Information Extraction*. PhD thesis, University of Washington, 1997. (Cited on pages 2 and 17.)
- [22] Ashwin Machanavajjhala, Arun Shankar Iyer, Philip Bohannon, and Srujana Merugu. Collective Extraction from Heterogeneous Web Lists. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 445–454, 2011. (Cited on pages 2 and 17.)
- [23] Bhaskara Marthi, Brian Milch, and Stuart Russell. First-order Probabilistic Models for Information Extraction. In *IJCAI 2003 Workshop on Learning Statistical Models from Relational Data*, 2003. (Cited on pages 2 and 17.)
- [24] Nandapandula Nakashole, Martin Theobald, and Gerhard Weikum. Scalable Knowledge Harvesting with High Precision and High Recall. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 227–236, 2011. (Cited on pages 2 and 17.)
- [25] Feng Niu, Ce Zhang, Christopher Ré, and Jude W Shavlik. DeepDive: Web-scale Knowledge-base Construction Using Statistical Learning and Inference. *2nd International Workshop on Searching and Integrating New Web Data Sources*, pages 25–28, 2012. (Cited on pages 2 and 17.)

- [26] Sebastian Riedel, Limin Yao, and Andrew McCallum. Modeling Relations and their Mentions without Labeled Text. In *Machine Learning and Knowledge Discovery in Databases*, pages 148–163. 2010. (Cited on pages 2 and 17.)
- [27] Sunita Sarawagi. Information Extraction. *Foundations and Trends in Databases*, 1(3):261–377, 2008. (Cited on pages 2, 13, and 17.)
- [28] Fabian M Suchanek, Mauro Sozio, and Gerhard Weikum. SOFIE: A Self-Organizing Framework for Information Extraction. In *Proceedings of the 18th International Conference on World Wide Web*, pages 631–640, 2009. (Cited on pages 2, 16, and 17.)
- [29] Petros Venetis, Alon Halevy, Jayant Madhavan, Marius Paşca, Warren Shen, Fei Wu, Gengxin Miao, and Chung Wu. Recovering Semantics of Tables on the Web. *Proceedings of the VLDB Endowment*, 4(9):528–538, 2011. (Cited on pages 2 and 17.)
- [30] Gerhard Weikum and Martin Theobald. From Information to Knowledge: Harvesting Entities and Relationships from Web Sources. In *Proceedings of the 29th ACM SIGMOD-SIGACT-SIGART symposium on Principles of Database Systems*, pages 65–76, 2010. (Cited on pages 2 and 17.)
- [31] Jun Zhu, Zaiqing Nie, Xiaojiang Liu, Bo Zhang, and Ji-Rong Wen. Statsnowball: A Statistical Approach to Extracting Entity Relationships. In *Proceedings of the 18th International Conference on World Wide Web*, pages 101–110, 2009. (Cited on pages 2 and 17.)
- [32] Guillermo Garrido, Anselmo Penas, Bernardo Cabaleiro, and Alvaro Rodrigo. Temporally Anchored Relation Extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 107–116, 2012. (Cited on pages 2 and 17.)
- [33] Erdal Kuzey and Gerhard Weikum. Extraction of Temporal Facts and Events from Wikipedia. In *Proceedings of the 2nd Temporal Web Analytics Workshop*, pages 25–32, 2012. (Cited on pages 2, 17, 26, and 79.)
- [34] Xiao Ling and Daniel S Weld. Temporal Information Extraction. In *Proceedings of the 24th AAAI Conference on Artificial Intelligence*, volume 10, pages 1385–1390, 2010. (Cited on pages 2, 17, 27, and 81.)
- [35] Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. Coupled Temporal Scoping of Relational Facts. In *Proceedings of the 5th ACM International*

- Conference on Web Search and Data Mining*, pages 73–82, 2012. (Cited on pages 2, 17, 81, and 101.)
- [36] Partha Pratim Talukdar, Derry Wijaya, and Tom Mitchell. Acquiring Temporal Constraints Between Relations. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, pages 992–1001, 2012. (Cited on pages 2 and 17.)
- [37] Marc Verhagen and James Pustejovsky. Temporal Processing with the TARSQI Toolkit. In *22nd International Conference on Computational Linguistics*, pages 189–192, 2008. (Cited on pages 2 and 27.)
- [38] Yafang Wang, Mingjie Zhu, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. Timely Yago: Harvesting, Querying, and Visualizing Temporal Knowledge from Wikipedia. In *Proceedings of the 13th International Conference on Extending Database Technology*, pages 697–700, 2010. (Cited on pages 2 and 17.)
- [39] Yafang Wang, Bin Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. Harvesting Facts from Textual Web Sources by Constrained Label Propagation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 837–846, 2011. (Cited on pages 2, 15, and 17.)
- [40] Yafang Wang, Maximilian Dylla, Marc Spaniol, and Gerhard Weikum. Coupling Label Propagation and Constraints for Temporal Fact Extraction. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 233–237, 2012. (Cited on pages 2, 17, 81, and 101.)
- [41] Erdal Kuzey, Jilles Vreeken, and Gerhard Weikum. A Fresh Look on Knowledge Bases: Distilling Named Events from News. In *Proceedings of the 23rd ACM International Conference on Information and Knowledge Management*, pages 1689–1698, 2014. (Cited on pages 5 and 78.)
- [42] Erdal Kuzey and Gerhard Weikum. EVIN: Building a Knowledge Base of Events. In *Proceedings of the 23rd International Conference on World Wide Web, Companion Volume*, pages 103–106, 2014. (Cited on page 5.)
- [43] Erdal Kuzey, Vinay Setty, Jannik Strötgen, and Gerhard Weikum. As Time Goes By: Comprehensive Tagging of Textual Phrases with Temporal Scopes. In *Proceedings of the 25th International Conference on World Wide Web*, pages 915–925, 2016. (Cited on page 5.)

- [44] Erdal Kuzey, Jannik Strötgen, Vinay Setty, and Gerhard Weikum. Temponym Tagging: Temporal Scopes for Textual Phrases. In *Proceedings of the 25th International Conference on World Wide Web, Companion Volume*, pages 841–842, 2016. (Cited on page 5.)
- [45] Douglas B Lenat. CYC: A Large-scale Investment in Knowledge Infrastructure. *Communications of the ACM*, 38(11):33–38, 1995. (Cited on page 7.)
- [46] Christiane Fellbaum. *WordNet*. Wiley Online Library, 1998. (Cited on pages 7, 11, 31, and 87.)
- [47] Kent A Spackman, Keith E Campbell, and Roger A Côté. SNOMED RT: A Reference Terminology for Health Care. In *Proceedings of the AMIA Annual Fall Symposium*, page 640, 1997. (Cited on page 7.)
- [48] Steffen Staab and Rudi Studer. *Handbook on Ontologies*. Springer Science & Business Media, 2013. (Cited on pages 7 and 8.)
- [49] Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. Web-scale Information Extraction in KnowItAll:(Preliminary Results). In *Proceedings of the 13th International Conference on World Wide Web*, pages 100–110, 2004. (Cited on page 8.)
- [50] Hugo Liu and Push Singh. ConceptNet-a Practical Commonsense Reasoning Tool-Kit. *BT Technology Journal*, 22(4):211–226, 2004. (Cited on page 8.)
- [51] S Auer, C Bizer, G Kobilarov, J Lehmann, R Cyganiak, and Z Ives. DBpedia: A Nucleus for a Web of Open Data. In *6th International Semantic Web Conference (ISWC), 2nd Asian Semantic Web Conference (ASWC)*, pages 715–728, 2007. (Cited on pages 8, 17, 21, and 26.)
- [52] Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. Freebase: A Collaboratively Created Graph Database for Structuring Human Knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 1247–1250, 2008. (Cited on pages 8, 17, 21, 26, and 77.)
- [53] Simone Paolo Ponzetto and Michael Strube. Wikitaxonomy: A Large Scale Knowledge Resource. In *18th European Conference on Artificial Intelligence*, volume 178, pages 751–752, 2008. (Cited on page 8.)

- [54] Thomas Hofweber. Logic and Ontology. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Fall 2014 edition, 2014. (Cited on page 8.)
- [55] Fabian M Suchanek. Automated Construction and Growth of a Large Ontology. *PhDThesis. Saarbrücken University, Germany*, 2008. (Cited on page 8.)
- [56] Tom Gruber. Ontology. *Encyclopedia of Database Systems*, pages 1963–1965, 2009. (Cited on page 8.)
- [57] Angus Stevenson. *Oxford Dictionary Of English*. Oxford University Press, USA, 2010. (Cited on page 11.)
- [58] Ned Markosian. Time. In Edward N. Zalta, editor, *The Stanford Encyclopedia of Philosophy*. Spring 2014 edition, 2014. (Cited on page 11.)
- [59] Michael David Fisher, Dov M Gabbay, and Lluís Vila. *Handbook of Temporal Reasoning in Artificial Intelligence*, volume 1. Elsevier, 2005. (Cited on page 12.)
- [60] Stuart Jonathan Russell, Peter Norvig, John F Canny, Jitendra M Malik, and Douglas D Edwards. *Artificial Intelligence: A Modern Approach*, volume 2. Prentice hall Upper Saddle River, 2003. (Cited on page 12.)
- [61] Chris Welty, Richard Fikes, and Selene Makarios. A Reusable Ontology for Fluents in OWL. In *Proceedings of the 2006 Conference on Formal Ontology in Information Systems*, volume 150, pages 226–236, 2006. (Cited on page 12.)
- [62] Octavian Udrea, Diego Reforgiato Recupero, and VS Subrahmanian. Annotated Rdf. *ACM Transactions on Computational Logic*, 11(2):10, 2010. (Cited on page 12.)
- [63] Joanna Biega, Erdal Kuzey, and Fabian M. Suchanek. Inside YAGO2s: A Transparent Information Extraction Architecture. In *Proceedings of the 22nd International World Wide Web Conference, Companion Volume*, pages 325–328, 2013. (Cited on page 13.)
- [64] Fabian M. Suchanek, Johannes Hoffart, Erdal Kuzey, and Edwin Lewis-Kelham. YAGO2s: Modular High-Quality Information Extraction with an Application to Flight Planning. In *Datenbanksysteme für Business, Technologie und Web*, pages 515–518, 2013. (Cited on page 13.)

- [65] Thomas Rebele, Fabian M. Suchanek, Johannes Hoffart, Joanna Biega, Erdal Kuzey, and Gerhard Weikum. YAGO: A multilingual knowledge base from wikipedia, wordnet, and geonames. (Cited on page 13.)
- [66] Dan Jurafsky. *Speech & Language Processing*. Pearson Education India, 2000. (Cited on page 13.)
- [67] Gian Piero Zarri. Automatic Representation of the Semantic Relationships Corresponding to a French Surface Expression. In *Proceedings of the First Conference on Applied Natural Language Processing*, pages 143–147, 1983. (Cited on page 13.)
- [68] Roger C Schank and Robert P Abelson. *Scripts, Plans, Goals, and Understanding: An Inquiry into Human Knowledge Structures*. Psychology Press, 1977. (Cited on page 13.)
- [69] James R Cowie. Automatic Analysis of Descriptive Texts. In *Proceedings of the First Conference on Applied Natural Language Processing*, pages 117–123, 1983. (Cited on page 13.)
- [70] Beth M Sundheim. Overview of the 3rd Message Understanding Evaluation and Conference. In *Proceedings of the 3rd Conference on Message Understanding*, pages 3–16, 1991. (Cited on page 14.)
- [71] Beth M Sundheim and Nancy A Chinchor. Survey of the Message Understanding Conferences. In *Proceedings of the Workshop on Human Language Technology*, pages 56–60, 1993. (Cited on page 14.)
- [72] Ralph Grishman and Beth Sundheim. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466–471, 1996. (Cited on pages 14 and 15.)
- [73] Nancy A Chinchor. Overview of Muc-7/Met-2. In *Proceedings of the 7th Conference on Message Understanding*, 1998. (Cited on page 14.)
- [74] George R Doddington, Alexis Mitchell, Mark A Przybocki, Lance A Ramshaw, Stephanie Strassel, and Ralph M Weischedel. The Automatic Content Extraction (ACE) Program-Tasks, Data, and Evaluation. In *Proceedings of the 4th International Conference on Language Resources and Evaluation*, pages 837–840, 2004. (Cited on page 14.)

- [75] ACE. Annotation Guidelines for Entity Detection and Tracking (EDT), 2004. URL <http://www ldc.upenn.edu/Projects/ACE/docs/EnglishEDTV4-2-6.PDF>. (Cited on page 14.)
- [76] Ludwig Wittgenstein and Gertrude Elizabeth Margaret Anscombe. *Philosophical Investigations*, volume 255. Blackwell Oxford, 1958. (Cited on page 14.)
- [77] Saul A Kripke. *Naming and Necessity*. Springer, 1972. (Cited on page 14.)
- [78] Gottlob Frege. Über Sinn Und Bedeutung. *Zeitschrift für Philosophie und philosophische Kritik*, 100:25–50, 1892. (Cited on page 14.)
- [79] Bertrand Russell. On Denoting. *Mind*, 14(56):479–493, 1905. (Cited on page 14.)
- [80] David Nadeau and Satoshi Sekine. A Survey of Named Entity Recognition and Classification. *Linguisticae Investigationes*, 30(1):3–26, 2007. (Cited on page 15.)
- [81] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-local Information Into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, 2005. (Cited on pages 15, 29, 81, and 84.)
- [82] Charles Sutton and Andrew McCallum. An Introduction to Conditional Random Fields for Relational Learning. *Foundations and Trends in Machine Learning*, 4(4):267–373, 2012. (Cited on page 15.)
- [83] Johannes Hoffart. *Discovering and Disambiguating Named Entities in Text*. PhD thesis, Universität des Saarlandes Saarbrücken, 2015. (Cited on page 15.)
- [84] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenaу, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust Disambiguation of Named Entities in Text. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 782–792, 2011. (Cited on pages 15, 82, 91, and 93.)
- [85] Razvan C Bunescu and Marius Pasca. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *11th Conference of the European Chapter of the ACL*, volume 6, pages 9–16, 2006. (Cited on page 15.)



- [86] Silviu Cucerzan. Large-scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, volume 7, pages 708–716, 2007. (Cited on page 15.)
- [87] Rada Mihalcea and Andras Csomai. Wikify!: Linking Documents to Encyclopedic Knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, pages 233–242, 2007. (Cited on page 15.)
- [88] David Milne and Ian H Witten. Learning to Link with Wikipedia. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management*, pages 509–518, 2008. (Cited on page 15.)
- [89] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective Annotation of Wikipedia Entities in Web Text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 457–466, 2009. (Cited on page 15.)
- [90] Lev Ratinov, Dan Roth, Doug Downey, and Mike Anderson. Local and Global Algorithms for Disambiguation to Wikipedia. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1375–1384, 2011. (Cited on page 15.)
- [91] Yang Li, Chi Wang, Fangqiu Han, Jiawei Han, Dan Roth, and Xifeng Yan. Mining Evidences for Named Entity Disambiguation. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1070–1078, 2013. (Cited on page 15.)
- [92] Pablo N Mendes, Max Jakob, Andrés García-Silva, and Christian Bizer. DBpedia Spotlight: Shedding Light on the Web of Documents. In *Proceedings of the 7th International Conference on Semantic Systems*, pages 1–8, 2011. (Cited on page 15.)
- [93] Dat Ba Nguyen, Martin Theobald, and Gerhard Weikum. J-NERD: Joint Named Entity Recognition and Disambiguation with Rich Linguistic Features. *Transactions of the Association for Computational Linguistics*, 4:215–229, 2016. (Cited on page 15.)
- [94] Omar Rogelio Alonso. Temporal Information Retrieval. *Computer Science*, pages 1–155, 2008. (Cited on page 15.)

- [95] Omar Alonso, Jannik Strötgen, Ricardo Baeza-Yates, and Michael Gertz. Temporal Information Retrieval: Challenges and Opportunities. In *Temporal Web Analytics Workshop TAWAW 2011*, page 1, 2011. (Cited on pages 15, 71, and 81.)
- [96] James Pustejovsky, José M Castano, Robert Ingria, Roser Sauri, Robert J Gaizauskas, Andrea Setzer, Graham Katz, and Dragomir R Radev. Timeml: Robust Specification of Event and Temporal Expressions in Text. *5th International Workshop on Computational Semantics*, 3:28–34, 2003. (Cited on pages 16, 26, and 78.)
- [97] Jannik Strötgen. *Domain-sensitive Temporal Tagging for Event-centric Information Retrieval*. PhD thesis, Institute of Computer Science, Ruprecht-Karls-University Heidelberg, 2015. (Cited on pages 16 and 111.)
- [98] Jannik Strötgen and Michael Gertz. Heideltime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, 2010. (Cited on pages 16, 75, 78, 84, 103, 110, and 111.)
- [99] Marc Verhagen, Roser Sauri, Tommaso Caselli, and James Pustejovsky. SemEval-2010 Task 13: TempEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 57–62, 2010. (Cited on pages 16, 18, and 26.)
- [100] Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. Open Information Extraction for the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, volume 7, pages 2670–2676, 2007. (Cited on page 16.)
- [101] Marti A Hearst. Automatic Acquisition of Hyponyms from Large Text Corpora. In *Proceedings of the 14th Conference on Computational Linguistics*, pages 539–545, 1992. (Cited on page 17.)
- [102] Anish Das Sarma, Alpa Jain, and Cong Yu. Dynamic Relationship and Event Discovery. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 207–216, 2011. (Cited on pages 18, 22, and 27.)
- [103] Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense Subgraph Maintenance under Streaming Edge Weight Updates for Real-time Story Identification. *Proceedings of the VLDB Endowment*, pages 574–585, 2012. (Cited on pages 18, 22, 27, and 36.)

- [104] Quang Xuan Do, Wei Lu, and Dan Roth. Joint Inference for Event Timeline Construction. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 677–687, 2012. (Cited on pages 18, 27, and 78.)
- [105] Wei Lu and Dan Roth. Automatic Event Extraction with Structured Preference Modeling. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 1301–1306, 2012. (Cited on pages 18 and 27.)
- [106] Manoj K. Agarwal, Krithi Ramamritham, and Manish Bhide. Real Time Discovery of Dense Clusters in Highly Dynamic Graphs: Identifying Real World Events in Highly Dynamic Environments. *Proceedings of the VLDB Endowment*, pages 980–991, 2012. (Cited on pages 18 and 36.)
- [107] Dafna Shahaf and Carlos Guestrin. Connecting the Dots Between News Articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–632, 2010. (Cited on pages 18 and 78.)
- [108] Dingding Wang, Tao Li, and Mitsunori Ogihara. Generating Pictorial Storylines Via Minimum-Weight Connected Dominating Set Approximation in Multi-View Graphs. In *Proceedings of the 26th AAAI Conference on Artificial Intelligence*, pages 683–689, 2012. (Cited on pages 18, 27, 54, and 55.)
- [109] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary Timeline Summarization: A Balanced Optimization Framework via Iterative Substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754, 2011. (Cited on pages 18 and 81.)
- [110] Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. Tempeval-3: Evaluating Events, Time Expressions, and Temporal Relations. *arXiv preprint arXiv:1206.5333*, 2012. (Cited on pages 18 and 26.)
- [111] Jannik Strötgen and Michael Gertz. HeidelTime: High Quality Rule-Based Extraction and Normalization of Temporal Expressions. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 321–324, 2010. (Cited on page 18.)

- [112] Angel X Chang and Christopher D Manning. SUTime: A Library for Recognizing and Normalizing Time Expressions. In *The 8th International Conference on Language Resources and Evaluation*, pages 3735–3740, 2012. (Cited on pages [18](#), [78](#), and [110](#).)
- [113] David Carmel, Ming-Wei Chang, Evgeniy Gabrilovich, Bo-June (Paul) Hsu, and Kuansan Wang. ERD’14: Entity Recognition and Disambiguation Challenge. *SIGIR Forum*, 48(2):63–77, 2014. (Cited on pages [19](#), [75](#), and [81](#).)
- [114] Marco Cornolti, Paolo Ferragina, and Massimiliano Ciaramita. A Framework for Benchmarking Entity-annotation Systems. In *Proceedings of the 22nd International Conference on World Wide Web*, pages 249–260, 2013. (Cited on pages [19](#), [75](#), and [81](#).)
- [115] Wei Shen, Jianyong Wang, and Jiawei Han. Entity Linking with a Knowledge Base: Issues, Techniques, and Solutions. *IEEE Transactions on Knowledge and Data Engineering*, 27(2):443–460, 2015. (Cited on pages [19](#), [75](#), and [81](#).)
- [116] Dafna Shahaf and Carlos Guestrin. Connecting the Dots Between News Articles. In *Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 623–632, 2010. (Cited on pages [22](#) and [27](#).)
- [117] Marc Verhagen, Robert Gaizauskas, Frank Schilder, Mark Hepple, Graham Katz, and James Pustejovsky. SemEval-2007 Task 15: TempEval Temporal Relation Identification. In *Proceedings of the 4th International Workshop on Semantic Evaluations*, pages 75–80, 2007. (Cited on page [26](#).)
- [118] James F Allen. Maintaining Knowledge about Temporal Intervals. *Communications of the ACM*, 26(11):832–843, 1983. (Cited on page [27](#).)
- [119] Marc Verhagen and James Pustejovsky. The TARSQI Toolkit. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 2043–2048, 2012. (Cited on pages [27](#), [75](#), and [78](#).)
- [120] Steven Bethard and James H Martin. Identification of Event Mentions and Their Semantic Class. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 146–154, 2006. (Cited on pages [27](#) and [78](#).)

- [121] Zornitsa Kozareva and Eduard Hovy. Learning Temporal Information for States and Events. In *Proceedings of the IEEE 5th International Conference on Semantic Computing*, pages 424–429, 2011. (Cited on pages 27 and 78.)
- [122] Jakub Piskorski, Jenya Belayeva, and Martin Atkinson. Exploring the Usefulness of Cross-lingual Information Fusion for Refining Real-time News Event Extraction: A Preliminary Study. In *Proceedings of the International Conference Recent Advances in Natural Language Processing*, pages 210–217, 2011. (Cited on pages 27 and 78.)
- [123] Matteo Brucato, Leon Derczynski, Hector Llorens, Kalina Bontcheva, and Christian S Jensen. Recognising and Interpreting Named Temporal Expressions. In *International Conference on Recent Advances in Natural Language Processing*, pages 113–121, 2013. (Cited on page 27.)
- [124] Prateek Jindal and Dan Roth. Extraction of Events and Temporal Expressions from Clinical Narratives. *Journal of Biomedical Informatics*, 46:S13–S19, 2013. (Cited on pages 27 and 78.)
- [125] James Allan, Ron Papka, and Victor Lavrenko. On-line New Event Detection and Tracking. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 37–45, 1998. (Cited on page 27.)
- [126] Nathanael Chambers and Dan Jurafsky. Template-based Information Extraction without the Templates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 976–986, 2011. (Cited on page 27.)
- [127] Evgeniy Gabrilovich, Susan Dumais, and Eric Horvitz. Newsjunkie: Providing Personalized Newsfeeds via Analysis of Information Novelty. In *Proceedings of the 13th International Conference on World Wide Web*, pages 482–490, 2004. (Cited on page 27.)
- [128] Yiming Yang, Tom Pierce, and Jaime Carbonell. A Study of Retrospective and On-line Event Detection. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 28–36, 1998. (Cited on page 27.)
- [129] Rui Yan, Xiaojun Wan, Jahna Otterbacher, Liang Kong, Xiaoming Li, and Yan Zhang. Evolutionary Timeline Summarization: A Balanced Optimization

- Framework via Iterative Substitution. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 745–754, 2011. (Cited on page 27.)
- [130] Xiaohua Hu, Xiaodan Zhang, Caimei Lu, E. K. Park, and Xiaohua Zhou. Exploiting Wikipedia as External Knowledge for Document Clustering. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 389–396, 2009. (Cited on page 31.)
- [131] Pu Wang, Jian Hu, Hua-Jun Zeng, and Zheng Chen. Using Wikipedia Knowledge to Improve Text Classification. *Knowledge and Information Systems, Vol. 19, Issue 3*, pages 265–281, 2009. (Cited on page 31.)
- [132] Evgeniy Gabrilovich and Shaul Markovitch. Overcoming the Brittleness Bottleneck Using Wikipedia: Enhancing Text Categorization with Encyclopedic Knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence*, pages 1301–1306, 2006. (Cited on page 31.)
- [133] ChengXiang Zhai. *Statistical Language Models for Information Retrieval*. Morgan & Claypool Publishers, 2008. (Cited on page 32.)
- [134] Jay M Ponte and W Bruce Croft. A Language Modeling Approach to Information Retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281, 1998. (Cited on page 32.)
- [135] John Lafferty and Chengxiang Zhai. Document Language Models, Query Models, and Risk Minimization for Information Retrieval. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 111–119, 2001. (Cited on page 32.)
- [136] Solomon Kullback and Richard A Leibler. On Information and Sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951. (Cited on page 33.)
- [137] Yang Zhou, Hong Cheng, and Jeffrey Xu Yu. Graph Clustering Based on Structural/Attribute Similarities. *Proceedings of the VLDB Endowment*, pages 718–729, 2009. (Cited on page 36.)
- [138] Arlei Silva, Wagner Meira, Jr., and Mohammed J. Zaki. Mining Attribute-Structure Correlated Patterns in Large Attributed Graphs. *Proceedings of the VLDB Endowment*, pages 466–477, 2012. (Cited on page 36.)

- [139] George Karypis and Vipin Kumar. Multilevel Graph Partitioning Schemes. In *Proceedings of the 1995 International Conference on Parallel Processing*, pages 113–122, 1995. (Cited on pages [38](#), [54](#), and [55](#).)
- [140] Ilya Safro, Peter Sanders, and Christian Schulz. Advanced Coarsening Schemes for Graph Partitioning. *Symposium on Experimental Algorithms, LNCS Vol. 7276*, pages 1–24, 2012. (Cited on page [38](#).)
- [141] Ning Zhang, Yuanyuan Tian, and Jignesh M Patel. Discovery-driven Graph Summarization. In *Proceedings of the 26th International Conference on Data Engineering*, pages 880–891, 2010. (Cited on page [38](#).)
- [142] Andreas Thor, Philip Anderson, Louiqa Raschid, Saket Navlakha, Barna Saha, Samir Khuller, and Xiao-Ning Zhang. Link Prediction for Annotation Graphs Using Graph Summarization. In *Proceedings of the 10th International Semantic Web Conference*, pages 714–729, 2011. (Cited on page [38](#).)
- [143] Yuanyuan Tian, Richard A. Hankins, and Jignesh M. Patel. Efficient Aggregation for Graph Summarization. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 567–580, 2008. (Cited on page [38](#).)
- [144] James Franklin. *The Science of Conjecture: Evidence and Probability Before Pascal*. JHU Press, 2015. (Cited on page [41](#).)
- [145] David JC MacKay. *Information Theory, Inference and Learning Algorithms*. Cambridge University Press, 2003. (Cited on page [41](#).)
- [146] Christopher M. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2007. (Cited on page [41](#).)
- [147] Jorma Rissanen. A Universal Prior for Integers and Estimation by Minimum Description Length. *The Annals of Statistics*, pages 416–431, 1983. (Cited on page [41](#).)
- [148] Peter D. Grünwald. *The Minimum Description Length Principle*. MIT Press, 2007. (Cited on page [41](#).)
- [149] Chris S. Wallace and David L. Dowe. Minimum Message Length and Kolmogorov Complexity. *The Computer Journal*, 42(4):270–283, 1999. (Cited on page [41](#).)

- [150] Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer Science & Business Media, 2009. (Cited on page 41.)
- [151] Leman Akoglu, Duen Horng Chau, Christos Faloutsos, Nikolaj Tatti, Hanghang Tong, and Jilles Vreeken. Mining Connection Pathways for Marked Nodes in Large Graphs. In *Proceedings of the 13th SIAM International Conference on Data Mining*, pages 37–45, 2013. (Cited on page 42.)
- [152] Danai Koutra, U. Kang, Jilles Vreeken, and Christos Faloutsos. Summarizing and Understanding Large Graphs. *Statistical Analysis and Data Mining*, 8(3): 183–202, 2015. (Cited on page 42.)
- [153] Danai Koutra, U. Kang, Jilles Vreeken, and Christos Faloutsos. VOG: Summarizing and Understanding Large Graphs. In *Proceedings of the 14th SIAM International Conference on Data Mining*, pages 91–99, 2014. (Cited on page 42.)
- [154] Saket Navlakha, Rajeev Rastogi, and Nisheeth Shrivastava. Graph Summarization with Bounded Error. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, pages 419–432, 2008. (Cited on page 42.)
- [155] Scott Kirkpatrick and Mario P Vecchi. Optimization by Simulated Annealing. *Science*, 220(4598), 1983. (Cited on page 48.)
- [156] Robin Sibson. SLINK: An Optimally Efficient Algorithm for the Single-link Cluster Method. *The Computer Journal*, 16(1):30–34, 1973. (Cited on page 54.)
- [157] Daniel Defays. An Efficient Algorithm for a Complete Link Method. *The Computer Journal*, 20(4):364–366, 1977. (Cited on page 54.)
- [158] Robert R Sokal. A Statistical Method for Evaluating Systematic Relationships. *University of Kansas Science Bulletin*, 38:1409–1438, 1958. (Cited on pages 54 and 55.)
- [159] Marc Spaniol, Julien Masanès, and Ricardo Baeza-Yates. The 4th Temporal Web Analytics Workshop (Tempweb’14). In *Proceedings of the 23rd International Conference on World Wide Web, Companion Volume*, pages 863–864, 2014. (Cited on pages 71 and 81.)
- [160] Naushad UzZaman, Hector Llorens, James Allen, Leon Derczynski, Marc Verhagen, and James Pustejovsky. Tempeval-3: Evaluating Events, Time



- Expressions, and Temporal Relations. *arXiv preprint arXiv:1206.5333*, 2012. (Cited on pages 72, 78, and 106.)
- [161] Jochen L. Leidner. Toponym Resolution in Text: Annotation, Evaluation and Applications of Spatial Grounding. *SIGIR Forum*, 41(2):124–126, 2007. (Cited on pages 75 and 81.)
- [162] Michael D Lieberman and Hanan Samet. Adaptive Context Features for Toponym Resolution in Streaming News. In *Proceedings of the 35th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 731–740, 2012. (Cited on pages 75 and 81.)
- [163] Kenton Lee, Yoav Artzi, Jesse Dodge, and Luke Zettlemoyer. Context-Dependent Semantic Parsing for Time Expressions. In *Proceedings of the Conference of the Association for Computational Linguistics*, pages 1437–1447, 2014. (Cited on page 78.)
- [164] Roser Saurí, Robert Knippen, Marc Verhagen, and James Pustejovsky. Evita: A Robust Event Recognizer for QA Systems. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 700–707, 2005. (Cited on page 78.)
- [165] Albert Angel, Nikos Sarkas, Nick Koudas, and Divesh Srivastava. Dense Subgraph Maintenance under Streaming Edge Weight Updates for Real-time Story Identification. *Proceedings of the VLDB Endowment*, 5(6):574–585, 2012. (Cited on page 78.)
- [166] Anish Das Sarma, Alpa Jain, and Cong Yu. Dynamic Relationship and Event Discovery. In *Proceedings of the 4th ACM International Conference on Web Search and Data Mining*, pages 207–216, 2011. (Cited on page 78.)
- [167] Vivien Petras, Ray R Larson, and Michael Buckland. Time Period Directories: a Metadata Infrastructure for Placing Events in Temporal and Geographic Context. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital libraries*, pages 151–160, 2006. (Cited on page 81.)
- [168] Melanie Feinberg, Ruth Mostern, Susan Stone, and Michael Buckland. Application of Geographical Gazetteer Standards to Named Time Periods. Technical report, Technical report, Electronic Cultural Atlas Initiative, Berkeley, 2003. (Cited on page 81.)

- [169] Yafang Wang, Bin Yang, Lizhen Qu, Marc Spaniol, and Gerhard Weikum. Harvesting Facts from Textual Web Sources by Constrained Label Propagation. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, pages 837–846, 2011. (Cited on page 81.)
- [170] Klaus Berberich, Srikanta Bedathur, Omar Alonso, and Gerhard Weikum. A Language Modeling Approach for Temporal Information Needs. In *Proceedings of the 32Nd European Conference on Advances in Information Retrieval*, pages 13–25, 2010. (Cited on page 81.)
- [171] Po-Tzu Chang, Yen-Chieh Huang, Cheng-Lun Yang, Shou-De Lin, and Pu-Jen Cheng. Learning-based Time-sensitive Re-ranking for Web Search. In *Proceedings of the 35th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1102, 2012. (Cited on page 81.)
- [172] Xiaoyan Li and W Bruce Croft. Time-Based Language Models. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pages 469–475, 2003. (Cited on page 81.)
- [173] Donald Metzler, Rosie Jones, Fuchun Peng, and Ruiqiang Zhang. Improving Search Relevance for Implicitly Temporal Queries. In *Proceedings of the 32nd international ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 700–701, 2009. (Cited on page 81.)
- [174] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. Enhancing Document Snippets using Temporal Information. In *String Processing and Information Retrieval*, pages 26–31, 2011. (Cited on page 81.)
- [175] Dhruv Gupta and Klaus Berberich. Identifying Time Intervals of Interest to Queries. In *Proceedings of the 23rd ACM International Conference on Conference on Information and Knowledge Management*, pages 1835–1838, 2014. (Cited on page 81.)
- [176] Nattiya Kanhabua, Tu Ngoc Nguyen, and Wolfgang Nejdl. Learning to Detect Event-Related Queries for Web Search. In *Proceedings of the 24th International Conference on World Wide Web, Companion Volume*, pages 1339–1344, 2015. (Cited on page 81.)
- [177] Omar Alonso and Kartikay Khandelwal. Kondenzer: Exploration and Visualization of Archived Social Media. In *Proceedings of the 30th International Conference on Data Engineering*, pages 1202–1205, 2014. (Cited on page 81.)

- [178] Xin Wayne Zhao, Yanwei Guo, Rui Yan, Yulan He, and Xiaoming Li. Timeline Generation with Social Attention. In *Proceedings of the 36th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1061–1064, 2013. (Cited on page [81](#).)
- [179] Stewart Whiting, Joemon Jose, and Omar Alonso. Wikipedia As a Time Machine. In *Proceedings of the 23rd International World Wide Web Conference, Companion Volume*, pages 857–862, 2014. (Cited on page [81](#).)
- [180] Ndapandula Nakashole, Gerhard Weikum, and Fabian Suchanek. PATTY: A Taxonomy of Relational Patterns with Semantic Types. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 1135–1145, 2012. (Cited on pages [84](#) and [88](#).)
- [181] Pawet Mazur and Robert Dale. WikiWars: a New Corpus for Research on Temporal Expressions. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 913–922, 2010. (Cited on pages [100](#) and [112](#).)
- [182] Lawrence D Brown, T Tony Cai, and Anirban DasGupta. Interval Estimation for a Binomial Proportion. *Statistical Science*, pages 101–117, 2001. (Cited on page [102](#).)

# INDEX

## A

AIDA ..... 15  
Attribute distance ..... 30  
Automatic Content Extraction ..... 14

## C

Chaining ..... 37  
Coarsening algorithms ..... 46  
Content distance ..... 30

## D

DBpedia ..... 1  
Distance measure ..... 29  
Document model ..... 34

## E

Entity ..... 9  
Entity extraction ..... 2, 14  
Epistemology ..... 1  
Event extraction ..... 78  
EVIN ..... 27  
EVIN browser ..... 66

## F

Fact extraction ..... 2  
Features ..... 29  
Freebase ..... 1

## G

Global model ..... 97  
Grouping ..... 37

## H

HeidelTime ..... 16, 111

## I

Induction by compression ..... 41  
Information extraction ..... 13  
Integer linear programming ..... 5

## J

Joint disambiguation ..... 95  
Joint model ..... 96

## K

Knowledge ..... 1

Knowledge base . . . . . 1, 7  
Knowledge base construction . . . . . 1  
Knowledge graph . . . . . 10  
Knowledge harvesting . . . . . 13  
Kullback-Leibler divergence . . . . . 33

**L**

Language model . . . . . 32  
Lexico-syntactic pattern . . . . . 87  
Literal . . . . . 9  
Local model . . . . . 95

**M**

MDL . . . . . 41  
Message Understanding Conferences 14  
METIS . . . . . 55  
Minimum description length . . . . . 5, 41  
Multi-view attributed graph . . . . . 5, 36  
MVAG . . . . . 36  
MVAG coarsening . . . . . 38

**N**

Named entities . . . . . 14  
Named entity disambiguation . . . 15, 81  
Named entity recognition . . . . . 14, 81

**O**

Ontological event . . . . . 26  
Ontology . . . . . 8  
Open IE . . . . . 16  
Optimization model . . . . . 41

**P**

Precision . . . . . 53, 55

**R**

Recall . . . . . 54, 55  
Relatedness measure . . . . . 92  
Relation extraction . . . . . 16

**S**

Semantic labeling . . . . . 23  
Semantic types . . . . . 29, 31  
Similarity metric . . . . . 92  
Story mining . . . . . 27

**T**

Tempeval . . . . . 26  
Template filling . . . . . 13  
Temponym . . . . . 4, 78  
Temponym detection . . . . . 86  
Temponym tagger . . . . . 111  
Temporal distance . . . . . 30  
Temporal expression . . . . . 78  
Temporal information extraction . . . 15  
Temporal knowledge . . . . . 2  
Temporal ordering . . . . . 23  
Temporal tagging . . . . . 4

**W**

Wikidata . . . . . 1

Y

YAGO..... 1