Enhancing Knowledge Acquisition Systems with User Generated and Crowdsourced Resources

Dissertation

zur Erlangung des akademischen Grades

eines Doktors der Ingenieurwissenschaften

der Naturwissenshaftlich-Technischen Fakultäten II

— Physik und Mechatronik —

der Universität des Saarlandes

von

Fang Xu

Saarbrücken

2012

Tag der Kolloquiums:	28.11.2013
Dekanin/Dekan:	Prof. Dr. Christian Wagner
Mitglieder des	
Prfungsausschusses:	Prof. Dr. Chihao Xu
	Prof. Dr. Dietrich Klakow
	PD. Dr. Günter Neumann

Eidesstattliche Versicherung

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

Affidavit

I hereby swear in lieu of an oath that I have independently prepared this thesis and without using other aids than those stated. The data and concepts taken over from other sources or taken over indirectly are indicated citing the source. The thesis was not submitted so far either in Germany or in another country in the same or a similar form in a procedure for obtaining an academic title.

Saarbrücken, _____ _____

(Datum / Date) (Unterschrift / Signature)

谁言寸草心

报得三春晖

— To my parents

Abstract

This thesis is on leveraging knowledge acquisition systems with collaborative data and crowdsourcing work from internet. We propose two strategies and apply them for building effective entity linking and question answering (QA) systems.

The first strategy is on integrating an information extraction system with online collaborative knowledge bases, such as Wikipedia and Freebase. We construct a Cross-Lingual Entity Linking (CLEL) system to connect Chinese entities, such as people and locations, with corresponding English pages in Wikipedia.

The main focus is to break the language barrier between Chinese entities and the English KB, and to resolve the synonymy and polysemy of Chinese entities. To address those problems, we create a cross-lingual taxonomy and a Chinese knowledge base (KB). We investigate two methods of connecting the query representation with the KB representation. Based on our CLEL system participating in TAC KBP 2011 evaluation, we finally propose a simple and effective generative model, which achieved much better performance.

The second strategy is on creating annotation for QA systems with the help of crowdsourcing. Crowdsourcing is to distribute a task via internet and recruit a lot of people to complete it simultaneously. Various annotated data are required to train the data-driven statistical machine learning algorithms for underlying components in our QA system. This thesis demonstrates how to convert the annotation task into crowdsourcing micro-tasks, investigate different statistical methods for enhancing the quality of crowdsourced annotation, and finally use enhanced annotation to train learning to rank models for passage ranking algorithms for QA.

Kurzfassung

Gegenstand dieser Arbeit ist das Nutzbarmachen sowohl von Systemen zur Wissenerfassung als auch von kollaborativ erstellten Daten und Arbeit aus dem Internet. Es werden zwei Strategien vorgeschlagen, welche für die Erstellung effektiver Entity Linking (Disambiguierung von Entitätennamen) und Frage-Antwort Systeme eingesetzt werden.

Die erste Strategie ist, ein Informationsextraktions-System mit kollaborativ erstellten Online-Datenbanken zu integrieren. Wir entwickeln ein Cross-Linguales Entity Linking-System (CLEL), um chinesische Entitäten, wie etwa Personen und Orte, mit den entsprechenden Wikipediaseiten zu verknüpfen.

Das Hauptaugenmerk ist es, die Sprachbarriere zwischen chinesischen Entitäten und englischer Datenbank zu durchbrechen, und Synonymie und Polysemie der chinesischen Entitäten aufzulösen. Um diese Probleme anzugehen, erstellen wir eine cross-linguale Taxonomie und eine chinesische Datenbank. Wir untersuchen zwei Methoden, die Repräsentation der Anfrage und die Repräsentation der Datenbank zu verbinden. Schlielich stellen wir ein einfaches und effektives generatives Modell vor, das auf unserem System für die Teilnahme an der TAC KBP 2011 Evaluation basiert und eine erheblich bessere Performanz erreichte.

Die zweite Strategie ist, Annotationen für Frage-Antwort-Systeme mit Hilfe von ”Crowdsourcing” zu erstellen. ”Crowdsourcing” bedeutet, eine Aufgabe via Internet an eine große Menge an angeworbene Menschen zu verteilen, die diese simultan erledigen. Verschiedene annotierte Daten sind notwendig, um die datengetriebenen statistischen Lernalgorithmen zu trainieren, die unserem Frage-Antwort System zugrunde liegen. Wir

zeigen, wie die Annotationsaufgabe in Mikro-Aufgaben für das Crowdsourcing umgewandelt werden kann, wir untersuchen verschiedene statistische Methoden, um die Qualität der Annotation aus dem Crowdsourcing zu erweitern, und schließlich nutzen wir die erweiterte Annotation, um Modelle zum Lernen von Ranglisten von Textabschnitten zu trainieren.

Acknowledgements

Working towards my PhD in the Spoken Language Systems group(*Lehrstuhl Sprach- und Signalverarbeitung*) at Saarland University has been an unique experience for me.

First and foremost, I am indebted to my supervisor Professor Dietrich Klakow, not only for his continued support but also for giving me the opportunity to carry out this research. He provided me with invaluable guidance at each stage during the writing of my thesis. His keen and vigorous academic observation enlightened me not only for this dissertation but also for my future studies and works.

I would like to thank Dr. Matthew Lease from the University of Texas at Austin for inspiration of my work on crowdsourcing and support during my visiting in Austin.

I also have to thank my colleagues for discussions and suggestions on various topics over the years. Special thanks to the following friends who proofread the manuscript and provided invaluable feedback: Dr. Michael Wiegand, Dr. Grzegorz Chrupała, Benjamin Roth, Mittul Singh and Stefan Kazalski.

I would also like to thank Diana Schreyer and Claudia Verburg for their administrative help during my PhD, and our system admistrator Dietmar Kuhn for enormous help.

Of course, none of the research in this thesis would have been carried out without the financial support provided by the Deutsche Forschungsgemeinschaft and Deutscher Akademischer Austauschdienst as part of their scholarship programme for which I am especially grateful.

The TREC/TAC evaluations have been an invaluable resource of both data and discussion

and I am indebted to the organisers.

Contents

1. Introduction **1**

1.1. Motivation 1

1.2. Cross-Lingual Entity Linking with Wikipedia 2

1.3. Crowdsourcing Annotation for Question Answering 4

1.4. Contributions 6

1.5. Outline of the thesis 7

I. Entity Linking with Wikipedia **9**

2. Background on Entity Linking **13**

2.1. TREC Entity Linking Evaluation 13

2.1.1. Task Definition 14

2.1.2. System Evaluation 18

2.2. Approaches to Monolingual Entity Linking 21

2.2.1. Query Expansion 21

2.2.2. Candidate Generation 23

2.2.3. Candidate Ranking 24

2.2.4. NIL Detection and Clustering 25

2.3. Approaches to Cross Lingual Entity Linking 27

2.3.1. Cross-Document Coreference Approach 29

2.3.2. Deductive One-model-for-all-language Approach 31

2.3.3. Transliteration and Translation-based Approach 33

2.3.4. Interlingual Representation Approach . . . 35
2.4. Conclusion . . . 37

3. Experimental Methodology **39**
3.1. Wikipedia Linking Structure . . . 39
3.2. Knowledge Resources . . . 41
3.2.1. TAC Knowledge Base . . . 41
3.2.2. Collections of Source Documents . . . 44
3.2.3. Chinese Knowledge Base . . . 44
3.2.4. Wikipedia Processing Tools . . . 46
3.3. English Language Processing Tools . . . 47
3.4. Chinese-Language NLP Tools . . . 48

4. Cross-Lingual Entity Linking System **51**
4.1. Introduction . . . 51
4.2. Prepossessing . . . 54
4.2.1. Background Knowledge Extraction . . . 54
4.2.2. Document and Query Processing . . . 55
4.2.3. Acronym Expansion . . . 55
4.3. Document Retrieval . . . 56
4.4. Entity Clustering . . . 58
4.5. Comparison of Cross Lingual and Monolingual EL . . . 62
4.6. Results . . . 66
4.6.1. Cross-lingual Entity Linking . . . 66
4.6.2. Result Analysis . . . 68
4.7. A Simple Chinese Entity Linking Model . . . 70
4.7.1. Candidate Generation . . . 70
4.7.2. Generative Cross-Lingual Entity Linking Model . . . 72
4.7.3. Evaluation . . . 76
4.8. Conclusion . . . 77

II. Learning with Crowdsourced Annotations **79**

5. Background on Question Answering **83**

5.1. Early History of Question Answering 84

5.2. Question Answering at TREC 86

5.3. Brief Overview of *Alyssa* QA System 93

6. Crowdsourcing for Paragraph Acquisition and Selection for QA **97**

6.1. Introduction 97

6.2. Amazon Mechanical Turk 98

6.3. Experiment Design 103

6.3.1. TREC data sets 103

6.3.2. Data and Experiment Setup 106

6.3.3. Data Quality Control 107

6.4. Related Work 111

6.5. Conclusion 115

7. True Annotation Learning **117**

7.1. Introduction 117

7.2. Majority Voting 117

7.3. Naive Bayes 118

7.4. GLAD Model 120

7.5. Results 123

7.6. Conclusion 126

8. Learning to Rank Supporting Passages for List Questions **129**

8.1. Introduction 129

8.2. Features 130

8.2.1. Question Representation 130

8.2.2. BOW Ranking Features 131

8.2.3. Query Proximity Features 132

8.2.4. NE Proximity Features . . . 133
8.2.5. WordNet-based Features . . . 134
8.2.6. Web Popularity-based Features . . . 136
8.2.7. Web-based Kernel Function . . . 137
8.3. Ranking model . . . 138
8.3.1. Support Vector Regression . . . 140
8.3.2. Ranking SVM . . . 141
8.3.3. SVM^{map} . . . 142
8.4. Experimental Results . . . 143
8.4.1. Datasets . . . 144
8.4.2. Evaluation Measures . . . 145
8.4.3. Baselines . . . 145
8.4.4. Results . . . 147
8.4.5. Performances on Retrieved Documents . . . 149
8.5. Conclusions . . . 150

9. Conclusions **153**

9.1. Summary . . . 153
9.2. Future Work . . . 155

List of Figures

2.1. General Cross-Lingual Entity Linking System Architecture [41]. 28
2.2. Language-independent Knowledge Base for Entity Linking. 30
2.3. Example of Neighbour nodes for entity mentions and KB nodes. First we find contextual neighbours of a query mention in background document. We generate KB entry candidates for the query mention and neighbours, which formalize the supporting matrix $\{R_{ij}\}$. The KB Node candidates are displayed in the row for q. Then we build neighbour relations between KB candidates by referring to anchor linkings in Wikipedia. For example, we find the Wikipage on "`Sevilla`" linking to Wikipage "`Spain`", therefore "`Spain`" is a KB neighbour of "`Sevilla`". The Wikipage on "`Andalusia`" links to "`Sevilla`", so it is also a neighbour of "`Sevilla`" 38

3.1. Sample Knowledge Base Entry on "*Theodore Roberts*". The `wiki_title` "**Theodore_Roberts**" is the base name of URL of the Wikipedia page. The `type` of entry is **PER** (person). The KB `id` is **E0000003**. The `facts` class covers content from the Wikipedia infobox. The `wiki_text` is the stripped version of Wikipedia article. 42
3.2. Sample Wikipedia Source text of the page on "*Theodore Roberts*" included in the English Wikipedia dump. 43

4.1. The overall architecture of our Monolingual Entity Linking System. 53
4.2. Linear interpolation of retrieval models searching on Chinese Wikipedia. . . 58

4.3. The demonstration of single linkage criterion for cluster similarity used in HAC algorithms. 60

4.4. The assignment of clusters based on the distance between a document and each centroid. 61

4.5. Performance of TAC KBP 2011 CLEL Systems on English and Chinese queries. 69

4.6. New CLEL system architecture with new candidate generation and candidate ranking components. 71

5.1. The archtecture of *Alyssa* Question Answering System. 95

6.1. preview of a HIT on phrase translation at AMT. On the top left shows the time that a worker has spent on the HITs, and the right top presents the number and value of submitted HITs so far. Below is the detailed information of HITs including the requester, reward per HIT, number of HITs, working duration and required qualifications (MTurkers are from non-US location and achieve minimum HIT approval rate of 85%). The following pane demonstrates the main interaction interface of the HIT. The language survey provides the assessment of workers' language proficiency; MTurkers need to translate Chinese phrases into English. When they click on the input field, a excerpt (highlighted in frame) is shown as explanatory reference. 99

6.2. Search-Continue-RapidAccept-Accept-Preview (SCRAP) interaction model from Heymann and Garcia-Molina [36]. 100

6.3. Sample real-time HIT with one pair of question and passage. The matched answers are highlighted with underline. The task instruction is, "*Given a question and a passage, please judge whether the passage answers the question. If the passage answers the question, and the passage contains one answer or more, the correct response is 'Yes'; If not, the response is 'NO'. Please only refer to the passage, don't use common sense.*". 107

6.4. Individual MTurker's accuracy vs. numbers of passages they completed. Each cross stands for an MTurker. Its abscissa value is total number of passages he completed. Its ordinate value is his annotation accuracy. . . . 109

7.1. Causal structure of GLAD model for inferring the hidden variables including task difficulties β, true label X, and labeler accuracies α given the observed labels L. Only the shaded variables are observed. 121

7.2. Accuracies of the approaches on dataset C vs. number of labels per HIT. All experimental trials are performed over 100 random samplings of labelers for all passages. The majority voting only consider odd numbers of labelers. For GLAD model, We used Gaussian priors $(\mu = 0.0001, \sigma = 0.0001)$ for α, Gaussian priors $(\mu = 0.0001, \sigma = 0.0001)$ for β' and the $\mathbf{X}$ are initialized with 0.0001. 125

8.1. Learning-to-rank framework (taken from Liu [59]). 139

8.2. Comparison of SVM^{map} and baseline on the TREC 2004 set of 33 questions. 150

9.1. Learning with Crowd Annotation. 154

List of Tables

2.1. Context of the query "**AZ**" from different background documents. 15

3.1. Number of documents in source collections. 44
3.2. Example of POS- and NER-tagged Sentence from Stanford CoreNLP. . . . 47
3.3. the tagset proposed by Institute of Computational Linguistics at Peking University 50

4.1. Tokenized representations of Chinese text resulting from different segmentation strategies. 57
4.2. Performance (Micro-Average Accuracy) of **cross-lingual** EL strategy for Chinese queries on TAC 2011 **development** data. 64
4.3. Performance of **monolingual** EL strategy for Chinese queries on TAC 2011 **development** data. 64
4.4. Sample translations generated with different methods. **N/T** stands for "No Translation". 65
4.5. Performance of different system configurations on the 2011 cross-lingual entity linking **evaluation** data. 67
4.6. Performance of different system configurations on the 2011 cross-lingual entity linking **development** data. 67
4.7. Micro-averaged accuracy of Chinese and English entities on TAC 2011 **evaluation** data. 67
4.8. Micro-averaged accuracy of Chinese entities on TAC 2011 **development** data with retrieval models on different indexing units. 68

4.9. Micro-averaged accuracy of Chinese entities on TAC 2011 **evaluation** data with models on different indexing units. 69
4.10. Overall performance of cross-lingual entity linking systems in TAC KBP 2012. 70
4.11. Sample candidate entities for the query “Elizabeth”, and their popularity in Chinese and English Wikipedia. 74
4.12. Micro-averaged accuracy of the Generative EL Model strategy for Chinese queries on TAC 2011 **evaluation** data. 76
4.13. Clustering evaluation of generative model EL strategy for Chinese queries on TAC 2011 **evaluation** data. 77

5.1. Task definition of TREC/TAC QA tracks. 88

6.1. Example of Answer Patterns. 104
6.2. Support judgement of passages matched with answer patterns. 104
6.3. Comparison of Datasets exported from AMT. The annotation accuracy improves with the increasing of the approval rate and workers per HIT. . . . 110
6.4. Inter Annotation Agreement for the 2856 question-passage pairs for both Dataset A and B. 110

7.1. Accuracies of the approaches on dataset A and B with different annotation accuracies (AA). 124
7.2. Examples of the *question ID*, *passage ID* and *answer string* following with the *supporting passage* for the question “`What countries has the IFC financed projects in?`” 127

8.1. Overview of the datasets used in the supporting documents evaluation. Q: question, Doc: document, and P: passages. Each element contain 3 values: total number of relevant passages / maximum number of relevant passages for a question / minimum number of relevant passages for a question. . . . 147
8.2. Performance comparison of passage selection models 148

8.3. Performance of ranking models using different individual and incremental feature groups. 148
8.4. Overview of the data used in the retrieved documents evaluation. 149
8.5. Performance comparison on retrieved documents data. 150

1. Introduction

1.1. Motivation

Recent decades have witnessed an explosive growth of data and information via Internet. It is not trivial to acquire and validate consistent knowledge from those massive structured and unstructured online data. The development of machine learning and data mining methods makes it possible to automatically extract and arrange information and knowledge. Researchers have been focusing on developing ad-hoc statistical methods for specific tasks or large-scale information processing and knowledge learning systems for mining continuous and high-volume data streams.

There are many successful applications of statistically inspired methods. For example, information retrieval [62] and statistical machine translation [48] support search engines and translation services provided by *Google* and *Bing*. It is vital to produce large-scale training, validation and test sets for those applications. A large dataset of queries and corresponding ranked lists of documents is requisite for training and evaluating ranking models for many applications of information retrieval; parallel corpora play a crucial role for training data-driven machine translation models. The lack of annotated and structured data is a critical obstacle for exploiting and developing supervised and semi-supervised machine learning methods [1], but it is expensive and tedious to carry out monotonous and repetitive annotation work.

[1] Although unsupervised machine learning methods do not need annotation, their performance is generally worse than supervised and semi-supervised methods.

In this thesis we present two strategies for soliciting annotated data with the help of collaboration via Internet. The first straightforward strategy is to directly mine useful information from online collaborative knowledge repositories, such as *Wikipedia* [2], *DBPedia* [3] and *Freebase* [4]. The second *crowdsourcing* strategy is more flexible. The contribution from a large crowd of online annotators via online platforms makes it possible to create ad-hoc annotation and evaluation according to the requirement of specific tasks. We show the application of these strategies by integrating them with realistic knowledge acquisition systems — Cross-Lingual Entity Linking and Question Answering systems respectively.

1.2. Cross-Lingual Entity Linking with Wikipedia

The core concept of Web 2.0 is to make the Internet into "a platform for information sharing, interoperability, user-centred design and collaboration" [5] for web users. User-generated content is a key characteristic of Web 2.0, which encourages users to publish their own content and collaborate on public content creation. As one of the most successful products of Web 2.0, Wikipedia is the largest and most popular multilingual, collaborative encyclopedia on the web with contributions from volunteers around the world. On December 31, 2011, there were over 3.8 million articles in English and versions available in 283 languages.

Wikipedia provides reliable knowledge and taxonomy for entities, and establishes relations between entities with inherent anchor links between articles. Wikipedia are featured with a large number of attributes about various facets of entities or events. The structured content of Wikipedia in XML format is easily machine-readable, which makes it applicable for a wide range of applications in natural language processing (NLP) [31, 20], information retrieval (IR) [75, 84] and information extraction (IE) [70, 19]. The Knowledge Base Population

[2] http://www.wikipedia.org/
[3] http://wiki.dbpedia.org/
[4] http://www.freebase.com/
[5] http://http://en.wikipedia.org/wiki/Web_2.0

(KBP) [6] evaluations sponsored by TAC aim at promoting research on knowledge extraction and integration based on Knowledge Base (KB) derived from Wikipedia, which has been a popular subject for research recently.

One important goal of KBP is to automatically link entities found in articles with nodes/entities in KB, namely *Entity Linking* (EL). Given a query and a background document containing mentions of the query, EL asks for whether the query corresponds to an entry in the KB, then clustering entities not in the KB. In the *Cross-Lingual Entity Linking* (CLEL) task, the query and background document is in Chinese while the KB is presented in English.

Although Wikipedia consists of many versions in different languages, the version of Wikipedia pages in English still substantially exceeds other language versions judging from the volume of the pages and content. The asymmetry of cross-lingual knowledge makes it challenging for CLEL task to bridge the language barrier. To overcome the obstacle, we collect the cross-lingual taxonomy and anchor links from Wikipedia to build a Chinese counterpart of English KB. We also investigate two different methods of connecting the query representation with the KB representation.

Another problem for both monolingual and cross-lingual EL is to resolve synonymy and polysemy of query mentions.

- Synonymy means entities can be presented in different forms, such as abbreviations and nick names. For example, “Michael Jordan” (NBA basketball player) is often mentioned by ***MJ***, ***M.J.***, and ***Air Jordan***.
- Polysemy means that identical query mentions can refer to different entities. For example, the name “Michael Jordan” represents *Michael Jeffrey Jordan, basketball player* [7] in the context “***Michael Jordan*** *plays basketball in Chicago Bulls.*”; whereas it refers to *Michael I. Jordan* [8] in the context “*Learning in Graphical Models:* ***Michael***

[6] http://www.nist.gov/tac/
[7] http://en.wikipedia.org/wiki/Michael_Jordan
[8] http://en.wikipedia.org/wiki/Michael_I._Jordan

Jordan".

To address these problems for CLEL, we propose a system with new candidate entity generation and generative entity ranking components.

1.3. Crowdsourcing Annotation for Question Answering

Question Answering (QA) is a challenging sub-field of NLP, aiming to identify and present the exact answer of a question to users, by building systems that can automatically analyse and understand the questions formulated in a natural language, such as "*Where was Franz Kafka born?* " and "*What books did Franz Kafka author?* ".

Automatic QA systems save time and effort for searching exact answers to questions out of superfluous information in documents such as web pages returned by a search engine. A typical pipeline architecture of QA systems consists of question analysis to extract key phrases from natural language questions, document retrieval to retrieve question-related documents from large collections of documents, passage retrieval to pinpoint answer-bearing passages/sentences in documents and answer extraction to extract actual answers from passages.

Recent decades have seen great progress in QA with a drift from traditional database-based approaches to statistical-based approaches. Statistical-based approaches have now become the dominant paradigm in QA as evidenced in government evaluations like NIST TREC [9] and the remarkable QA supercomputer Watson [10] designed by IBM. Both the dominance of the statistical approach in QA and the progress made in recent years are clearly demonstrated in QA evaluations organized by NIST, Cross Language Evaluation Forum (CLEF) [11] and NTCIR (NII-NACSIS Test Collection for IR Systems) [12]. These initiatives have not only fuelled the overall research process in QA but have also led to the

[9] http://trec.nist.gov/data/qamain.html
[10] www.ibm.com/innovation/us/watson/index.html
[11] http://www.clef-initiative.eu/
[12] http://research.nii.ac.jp/ntcir/

development of successful statistical algorithms for various QA tasks.

The state-of-the-art approaches to QA are data-driven requiring a considerable amount of annotated corpora. To train various components in QA systems, various levels of annotations are required. The QA benchmarks at TREC only provides annotated pairs of answers and source documents. The lack of judgement of answer passages hinders the development of passage retrieval and answer extraction methods, which play key roles in QA and considerably influence the performance of QA systems.

We resort to crowdsourcing techniques [38] for building annotations of answer-bearing passages. Crowdsourcing services such as Amazon Mechanical Turk (AMT) [13] have made it easy to distribute those annotation tasks to a large crowd of online workers. With the large crowd of online workers, developers or researchers can submit their micro-tasks, get them done swiftly, approve or refuse completed tasks and combine the results into their own applications. We present a practical paradigm for learning with crowdsourcing annotation. It includes the following steps:

1. Convert an annotation task into a group of crowdsourcing micro-tasks.
2. Enhance the quality of crowdsourcing annotation with methods modeling the characteristics of annotators and annotation items.
3. Use enhanced annotation for training and testing on a variety of QA tasks.

We describe designing and running batch micro-tasks via AMT. As crowd annotations are unreliable and inconsistent, we investigate methods for learning true labels from noisy crowd annotation via major voting, naive Bayesian classification, and expectation maximization strategies. Finally those enhanced annotations are used to train and test passage ranking component for our QA system.

[13] http://www.mturk.com/

1.4. Contributions

The main contribution of my thesis is three-folds:

- **Cross-Lingual Entity Linking Systems**. This thesis contributes to the development of a cross-lingual entity linking (CLEL) system for the TAC evaluation. I was in charge of designing the overall architecture of the TAC CLEL system, as well as exploiting methods for module implementation. I implemented entity generation and clustering modules in the TAC system. I proposed a new generative entity ranking model for CLEL, which achieved significant increase in linking performance over the original system.

- **Crowdsourcing Annotation and Learning**. Experiments via AMT are run to annotate supporting passages for list question answering [14]. To improve the quality of crowd annotations, I compared three different models for learning the gold-standard annotation from noisy crowd annotations from the perspective of supervised and unsupervised learning. The comparable performances of those models showed the feasibility of applying crowdsourcing on collecting annotations and further improve the data quality with appropriate unsupervised methods.

- **Learning to Rank Passages for Question Answering**. It is important to determine the influence of crowdsourced annotations on the performance of QA system. Taking the passage ranking task as an example, we applied learning to rank models on ranking supporting passages training on the crowd-annotated list question datasets. Experimental results indicate that learning to rank models based on crowd annotations achieve start-of-the-art performance.

[14] List questions request a set of instances as answers.

1.5. Outline of the thesis

This thesis is divided into two main parts. The first part discusses CLEL with collaborative structured data from Wikipedia. It includes the following chapters:

- **Chapter 2** contains a general definition of the EL and CLEL problems. After reviewing the history of EL at TAC, we cover a broad overview of different monolingual and cross-lingual paradigms to give a big picture about the entire space of CLEL with Wikipedia. We systematically categorize and describe various kinds of underlying modules adopted in TAC participating systems.
- **Chapter 3** describes our work on extracting structured knowledge across different language versions of Wikipedia. Knowledge resources and natural processing tools for building the CLEL system are also listed for the reference in later chapters.
- **Chapter 4** presents the architecture of our two CLEL systems. Algorithms of each underlying modules are detailed. We thoroughly evaluate and analyse individual models and overall CLEL system. Following our TAC system, a CLEL system with new candidate generation and generative entity ranking modules are introduced to boost the performance.

The second part covers crowdsourced annotation and learning for QA.

- **Chapter 5** briefly describes the history of QA techniques and the development of international QA evaluations sponsored by NIST. We also present the detailed architecture of our latest QA system to show how passage ranking method works in overall QA system.
- **Chapter 6** introduces the infrastructure and facilities of online crowdsourcing platform AMT. We introduce the essentials of crowdsourcing annotation workflow including data processing, experimental interface design, and annotation collection and post-processing. We show the importance of controlling online annotation activities with built-in functions from AMT. Proper experimental settings can reduce a part of

noisy and incompetent annotators and thus decrease adverse influence on learning models introduced in Chapter 7.

- **Chapter 7** shows how to enhance the quality of annotation generated by online annotators. We implement three methods to improve annotation from noisy user-generated labels. Those methods infer the gold-standard annotation by capturing statistical characteristics of different annotators, annotation instances and true labels. Experimental results show that unsupervised machine learning methods can learn the true labels effectively.
- **Chapter 8** applies learning to rank methods to rank supporting passages for the list questions defined by TREC QA evaluations. Those models are trained on our enhanced annotation. To our knowledge, the passage ranking task for list questions has not been well explored in previous works.

To wrap up, **Chapter 9** summarizes the thesis by drawing overall conclusions and discussing possible research work in the future.

Part I.

Entity Linking with Wikipedia

Introduction

Named entity identification and disambiguation has been established as an important and fundamental task in NLP. Previous research focused on recognizing named entities from text, resolving ambiguous name entities and integrating them into other tasks such as machine translation, question answering and information retrieval. Recently with the proliferation of social collaboration and knowledge sharing websites such as *Wikipedia* [15] and *Quora* [16], research interests have gradually shifted to the discovery of rich knowledge and properties of named entities from those large-scale Knowledge Bases (KBs). Text Analysis Conference (TAC) proposed the Knowledge Base Population (KBP) track to retrieve and collect distributed information about an entity that contained with large document collections, and extracts the information to populate an existing KB [41]. TAC derived the KB from the largest online collaborative knowledge repository — Wikipedia, and proposed task definition, experiment data and evaluation measure.

KBP track includes two main shared tasks: *Entity Linking* (EL) and *Slot Filling* (SF). The input for EL comprises a query name string and a document containing the query. EL asks whether the query corresponds to an entry in the KB [42], then clusters the entities not included in the KB [41]. The requirement for the SF systems is to find the values of specified attributes ("slot") of the entity from a large collection of source documents, such as the birthday and children of a person or the website and employees of an organization. A query for SF contains a name-string, document ID, entity-type, node-id (entry ID) in Wikipedia, and an optional list of slots to ignore. The main goal of KBP tracks is to bridge

[15] http://www.wikipedia.org
[16] http://www.quora.com/

the information extraction and question answering communities and promote research in exploring facts about entities and broadening a knowledge based with these facts [41] [17].

This part of my thesis studies the novel Cross-Lingual EL (CLEL) task proposed in KBP 2011. The query for CLEL is bilingual, presented in either English or Chinese. We provide an overview of monolingual EL and CLEL tasks. We detail the implementation of our CLEL system participating in TAC 2011 evaluation and new experiments of a generative entity ranking model.

Refer to [17, 115] for detailed information about our participation in SF tasks.

[17] TAC KBP 2012 will have a new task –*Cold Start Knowledge Base Population*: Given a KB schema with an empty knowledge base, build the KB from scratch by mining a large text collection.

2. Background on Entity Linking

2.1. TREC Entity Linking Evaluation

Since 2008, NIST (National Institute of Standards and Technology) has been organizing the annual TAC and holding a series of tracks (*a.k.a.* shared tasks) for large-scale evaluation of NLP methodologies. Those evaluation tracks include Question Answering, Text Summarization, Recognizing Textual Entailment and Knowledge Base Population. Each year TAC also adjusts and arranges the tasks according to previous tracks and new track proposals.

The annual circulation of a TAC track starts with the announcement of task description and call for participations. Then during the *developing period*, TAC releases the source collections and labelled development dataset so all participants can develop and tune ad hoc systems. Then during the one-week *evaluation period*, participants receive the unlabelled evaluation dataset from TAC and submit system-generated results, which will be evaluated by TAC. Eventually the conference is held for *retrospective and prospective studies* on all tracks so that participants can discuss about ideas and share opinions of future TAC evaluation.

Knowledge Base Population (KBP) track was first introduced in TAC 2009 and has been running with a variety of shared tasks for three years. As we mentioned before, the goal of KBP is to explore facts about entities in a large corpus and use them to enrich the knowledge base (KB). The first step of KBP is to determine whether an entity exists in KB then make sure it is correctly linked, namely the *Entity Linking* (EL) task. For those

entities out of KB, we can harvest and pinpoint required attributes related to it with the help of *Slotting Filling* system. Eventually we can extend the KB with arrangement and construction of new entities and their featured slots.

Although the KB is in English, the Cross-Lingual Entity Linking (CLEL) system can help crossing the language barrier and mining multilingual knowledge, such as Chinese and Spanish. We emphasize on CLEL in this thesis, and we will provide detailed instructions on the development of evaluation benchmarks, and consequently the adjustment of evaluation metrics.

2.1.1. Task Definition

The EL task provides several queries, each of which contains a name string, which is a named entity about a person(PER), organization(ORG) or geo-political (GPE, a location about a government), and a background document ID. The document provides background information about the query. Given a query q and a set of name mentions $M = m_1, m_2, \ldots, m_k$ mined from documents KB, the objective is to find a set of entities $E = e_1, e_2, \ldots, e_n$. Each participating system shall return the ID of a KB entry to which the name mention refers to or NIL if no such KB entry is found.

The English EL tasks have been sponsored since 2009. Sample queries are listed as following.

```
<query id="EL000014">
  <name>AZ</name>
  <docid>eng-WL-11-174595-12967314</docid>
</query>

<query id="EL000019">
  <name>AZ</name>
  <docid>eng-WL-11-174646-13000609</docid>
</query>

<query id="EL000026">
  <name>AZ</name>
```

```
  <docid>eng-WL-11-174574-12934438</docid>
</query>

<query id="EL000029">
  <name>AZ</name>
  <docid>eng-WL-11-174595-12967466</docid>
</query>
```

Listing 2.1. Sample English Queries in KBP Entity Linking.

Document ID	Context
eng-WL-11-174595-12967314	*. . . left Scottsdale,* ***Arizona*** *and are now back home in LA. . . . they might purchase before they left Scottsdale,* ***AZ****:*
eng-WL-11-174646-13000609	*Shelley Farringer is back in* ***AZ*** *Saturday . . .*
eng-WL-11-174574-12934438	*Here she is in a new independent film alongside the likes of Ray J, LisaRaye,* ***AZ*** *and more.*
eng-WL-11-174595-12967466	*Stefan, a DJ at The Zone 101.5 FM in Phoenix,* ***AZ****, sent me an awesome MP3 of the interview. . .*

Table 2.1.. Context of the query "**AZ**" from different background documents.

Given those queries in the Listing 2.1, it is impossible to map them to the corresponding entry due to lack of evidence. Those contexts from background documents in Table 2.1 therefore provides more information to help identify and disambiguate those queries.

According to the gold standard annotation, the queries "EL000014", "EL000019" and "EL000029" shall be linked to the KB entry with ID "E0690220" (titled "Arizona" [1]); the query "EL000026" shall be linked to the KB entry "E0206158" ("Anthony Cruz" [2]). This example shows that main challenges in EL include *synonymy*, i.e., multiply candidates for acronyms and popular names (e.g., Arizona for "AZ"); and *polysemy*, i.e., a name string refers to different entities.

In 2010, TAC made minor change of the source collection by introducing a large amount of web documents for regular EL tasks, and introduced an optional EL task, which prevent teams from using the free text ("wiki_text") in each KB entry for building EL systems. Only

[1] http://en.wikipedia.org/wiki/Arizona
[2] http://en.wikipedia.org/wiki/AZ_(rapper)

the attributes extracted from infoboxes of KB entries can be used for linking queries [42].

KBP2011 run both the English EL tasks and a new *CLEL* task, in which the queries are bilingual, in either English or Chinese while the KB is in English. Some queries are denoted in the Listing 2.2.

```
<query id="EL_CLCMN_03011">
  <name>李娜</name>
  <docid>XIN_CMN_20050429.0146</docid>
</query>

<query id="EL_CLCMN_03012">
  <name>李娜</name>
  <docid>XIN_CMN_20080211.0135</docid>
</query>

<query id="EL_CLCMN_03013">
  <name>李娜</name>
  <docid>XIN_CMN_19991116.0016</docid>
</query>

<query id="EL_CLENG_03959">
  <name>Li Na</name>
  <docid>AFP_ENG_20070116.0014.LDC2009T13</docid>
</query>
```

Listing 2.2. Sample Cross Lingual Queries in KBP Entity Linking.

An ideal CLEL system shall learn that both of the queries "EL_CLCMN_03012" and "EL_CLCMN_03011" refer to the KB entry "E00026964" (the track cyclist named Li Na) [3], while cluster the remaining queries and connect them with another KB entry "E0128750" (the tennis player) [4].

Another alteration in 2011 is that NIL queries shall be clustered into different sense (topic) groups, each of which refers to different entities even they are not included in the KB. Some disambiguate queries are shown in Listing 2.3.

[3] http://en.wikipedia.org/wiki/Li_Na_(cyclist)
[4] http://en.wikipedia.org/wiki/Li_Na_(tennis)

```
<query id="EL_CLCMN_02405">
  <name>华莱士</name>
  <docid>AFP_CMN_20060326.0013</docid>
</query>

<query id="EL_CLCMN_02406">
  <name>华莱士</name>
  <docid>AFP_CMN_20070331.0025</docid>
</query>

<query id="EL_CLCMN_02411">
  <name>华莱士</name>
  <docid>PDA_CMN_20061113.0268</docid>
</query>

<query id="EL_CLENG_04292">
  <name>Wallace</name>
  <docid>APW_ENG_20070201.1229.LDC2009T13</docid>
</query>

<query id="EL_CLENG_04294">
  <name>Wallace</name>
  <docid>APW_ENG_20070330.1282.LDC2009T13</docid>
</query>
```

Listing 2.3. Cross Lingual Queries with linked and NIL References.

All queries literally mean the person name "华莱士" ("Wallace"), however they are connected with different referrals. Only query "EL_CLCMN_02411" is linked to KB entry "E0568129"("William Wallace" [5]), while other entities have no KB referent and hence are tagged as "NIL". Regarding the NIL clustering task, the "EL_CLCMN_02406" and "EL_CLENG_04294" are clustered together as they are reporting on "Editor Richard Wallace of British Daily Mirror newspaper", while "EL_CLCMN_02405" and "EL_CLENG_04292" are in another cluster on "Liberia's Foreign Minister George Wallace". The number of senses is not known in advance, which makes it harder than the Word Sense Disambiguation (WSD) problem.

The prime issues involved with EL tasks can be summarized as following. First, queries

[5] http://en.wikipedia.org/wiki/William_Wallace

are presented in different variations, i.e., different query strings can refer to the same entity. On the other hand, identical query strings are involved with different references and bring ambiguities in query meanings. While dealing with the *synonymy and polysemy* issues, a system also need make decisions on choosing optimal cut-off scores for determining NIL entities and clustering them into proper groups. The cross-lingual task raises the challenging problems of query translation and cross-lingual information extraction. To address those problems, Section 2.2 and 2.3 will review typical approaches used for monolingual and cross-lingual EL tasks. In Chapter 4, we will introduce our system participating in KBP 2011 CLEL tasks and thoroughly discuss the components in the system.

2.1.2. System Evaluation

The effectiveness of Entity Linking (EL) system can be evaluated by several measures. For each singular query, the KB entry ID (or NIL) generated by an EL system is checked against the gold standard to see whether correctly linked or not. To judge the overall performance, TAC 2009 proposed Micro-Average Accuracy (MicroAcc) to evaluate the effectiveness of an EL system over all queries, and Macro-Average Accuracy based on individual topics (referred entries).

Micro-Average Accuracy is defined as the fraction of the generated identifiers which are equal to the gold standard, i.e.,

$$\text{Accuracy}_{\text{micro}} = \frac{\text{NumCorrect}}{\text{NumQueries}} \tag{2.1}$$

Here, $NumQueries$ is the number of input queries. $NumCorrect$ is the number of correctly linked queries. Each NIL query is regarded as an unique query.

Macro-Average Accuracy (MacroAcc) is the simple average of Micro-Average Accuracies on individual topics, each of which gets the same weight in the average. i.e.,

$$\begin{aligned} \text{Accuracy}_{\text{macro}} &= \frac{\sum_i^{\text{NumEntities}} \text{MicroAcc}_i}{\text{NumEntities}} \\ &= \frac{\sum_i^{\text{NumEntities}} \frac{\text{NumCorrect}(E_i)}{\text{NumQueries}(E_i)}}{\text{NumEntities}} \end{aligned} \tag{2.2}$$

Here, $NumEntities$ denotes the number of unique referred KB entries. Given the linking information from the gold standard data, we first measure the individual MicroAcc$_i$ on each reference entry E_i independently, then the average of MicroAcc$_i$ over all referred entries is obtained as the final MacroAcc. Since certain entries are harder to determine, or relate to more queries, *MacroAcc* can score the influence of topic variation in evaluation data.

KBP tracks before 2011 did not demand EL systems to cluster NIL answers. NIL queries domains almost half of all queries. The MacroAcc is less useful for evaluating the NIL accuracy. Therefore only MicroAcc was used as the official score in the first two EL evaluations.

Due to the requirement of clustering NIL nodes according to their reference, TAC modified the metrics for the evaluation of performance of new NIL clustering task in KBP 2011. Instead of counting on individual queries, the performance is based on aggregated correctness of query clusters. They evaluate the clustering results with the scoring metric — B–Cubed$+$, defined as following.

Let $L(e)$ and $C(e)$ be the topic (gold-standard cluster) and the system-generated cluster of an entity mention e, while the system and gold-standard KB identifier for an entity mention e is notated as $SI(e)$ and $GI(e)$.

If two entities mentions clustering into the same topic also share the same KB identifier as the gold-standard, they are considered to be correctly related. The correct relatedness $G(e, e')$ between two entity mention e and e' is calibrated with the following formula:

$$G(e,e') = \begin{cases} 1 & \text{iff } L(e) = L(e') \wedge C(e) = C(e') \wedge \\ & \quad GI(e) = SI(e) = GI(e') = SI(e') \\ 0 & \text{otherwise} \end{cases} \tag{2.3}$$

The effectiveness of EL system is assessed by *precision* and *recall*. The B-cubed+ Precision is denoted in formula 2.4. Given a mention e, for each e' sharing the cluster with e (denoted as $C(e) = C(e')$), the correct relatedness of e' is judged by using the distribution 2.3. The precision regarding e, denoted as $\text{Avg}_{e'.C(e)=C(e')}[G(e,e')]$ is the proportion of correctly linked entity mentions e' within the cluster. Eventually the overall B-Cubed+ precision is the average of precisions from all mentions e. Similarly the B-Cubed+ Recall described in formula 2.5 is defined on entities appearing in a topic (denoted as $L(e) = L(e')$) instead of a cluster.

$$\text{Precision} = \text{Avg}_e \left[\text{Avg}_{e'.C(e)=C(e')} \left[G(e,e')\right]\right] \tag{2.4}$$

$$\text{Recall} = \text{Avg}_e \left[\text{Avg}_{e'.L(e)=L(e')} \left[G(e,e')\right]\right] \tag{2.5}$$

$$\text{F-measure} = \frac{2 \times \text{Precision} \times \text{Recall}}{(\text{Precision} + \text{Recall})} \tag{2.6}$$

There is a trade-off between precision and recall: when precision increases, recall usually decreases, and vice visa. To give even weight to the recall and precision, their harmonic mean — F-measure (Formula 2.6) is the official scoring. The KBP2011 scores evaluated with B-cubed+ F-Measure highly correlate with the scores based on the traditional Micro-Average Accuracy metric used in KBP 2009 and 2010. [41]

2.2. Approaches to Monolingual Entity Linking

All the EL systems in TAC adopted the pipeline architecture, i.e., performing a succession of different components with queries as the initial input to reach the final decision. The final accuracy of an system is determined by the soundness of its underlying components. Ji et al. [41] summarized the typical approaches featured in evaluation systems of TAC 2011. In this section, we will thoroughly describe representative underlying components embedded in successful mono-lingual EL systems.

The architecture of monolingual EL system typically consists of the following components:

1. **Query Expansion** expands and reformulates the query into a set of richer forms by mining anchor and link information from Wikipedia or possible co-references from the background documents.
2. **Candidate Generation** produces all KB entries to which a query possibly refers based on the result of query expansion.
3. **Candidate Ranking** ranks all the KB candidates by estimating their relevance to the query.
4. **NIL Detection and Clustering** determines KB entries from Step 3 with the linking confidence than a threshold as NIL, then clusters all those NIL queries.

2.2.1. Query Expansion

Regarding sample queries shown in Listing 2.1, it is hard to pinpoint the correct entity from KB without any expansion of the queries. Two critical problems need to be solved to ensure following components achieving the optimal performance.

- the main problem is to expand *acronyms* and *abbreviations*, such as “UT” stands for “University of Texas” or “University of Tennessee”. Especially *alias* and *nicknames* are not trivial to resolve, e.g., for the query “iron lady”, 16 leaders earned the unofficial

titles. [6]

- Systems shall discover supportive and complementary evidence for the query string. For example, given the query "London", possible KB entry candidates include `E0001618`: "London, Kentucky", `E0026729`: "London, Ontario", `E0104817`: "London, Arkansas", `E0104817`: "London (novel)", and `E0397283`: "London". Additional evidence is demanded to distinguish them.

Query expansion approaches aim at mining the background documents [72, 54, 12] and Wikipedia structural information [72, 54, 97, 12, 124]. The best EL system of TAC2009 [97] used Wikipedia links and titles for query expansion. They utilized redirect pages as the synonym of entities and disambiguation pages for homonym resolutions. The LCC system [72, 54] created Wikipedia knowledge repository of approximately 28 million terms for both query expansion and candidate generation by collecting normalized articles and redirect titles, creating a dictionary of surface texts of hyperlink anchors to targets and a dictionary of each disambiguation page title to its including titles, therefore if the query string matches with any key of any dictionaries, the corresponding values are possible expansions. The LCC system also leveraged information from the source document contexts, including:

- *longer mention* of the entity query string (e.g., query "Black Panther" $\rightarrow$ phrase "New Black Panther");
- *soft mention* with measuring word matching(e.g., mention "Moss" $\rightarrow$ named entity "Carrie Ann Moss");
- *contextual mention* of the acronym(e,g, "in the Democratic Republic of Congo (**DCR**)")
- *retrieved Wikipedia pages* with the query string with the help of search engines like Google.

[6] http://en.wikipedia.org/wiki/Iron_lady

Moreover, some systems use name entity recognition tools to tokenize and extract named entities (NEs) from documents. Those NEs containing the query string are considered as expansion [80, 12].

Besides those surface matching heuristics, some systems adopted statistic-based methods for expanding name variations. NUSchime system [124] selected the correct acronym expansions with an supervised classifier trained on various features. NLPR systems [34, 122] extracted the entity candidates for abbreviation queries from Wikipedia texts by using regular expression patterns to match terms with captain initialization.

Most EL systems follow those strategies of expanding queries, and accordingly generate KB candidates.

2.2.2. Candidate Generation

The goal of Query Expansion is to discover as much information of a query as possible, therefore to boost the potential recall of entity linking performance. The Candidate Generation procedure shares the same purpose, meanwhile emphasizes on producing reasonable size of candidate sets without introducing too many irrelevant KB entries. Those procedures highly relate together and often take place jointly or simultaneously, such as those in LCC systems. [72, 54]

Surface form matching is widely used to select KB candidates. A common strategy is to choose KB titles with exact or approximate string matching of the query string [67, 1]. The query strings are matched against the following Wikipedia information to determine KB entry candidates.

- Redirect and disambiguation pages are extracted to create dictionaries of acronyms and aliases to KB entries [97, 98, 67, 55, 80, 123].
- Mapping of Wikipedia hyperlink anchor texts to KB titles [34, 55] (e,g, both “IBM” and “Big Blue” link to the entry “IBM”).

- Calculating edit distances between query strings and titles [90].
- Bold texts at the beginning of Wikipedia text is considered as alias as well [98] (e.g., "**International Business Machines Corporation** " in the Wikipage on "IBM" [7]).
- Wikipages retrieved by search engines [34, 54, 1].

The candidate generation problem can be converted to an information retrieval (IR) problem on searching the KB collection with a query and its expansions. Bikel et al. [9] applied fuzzy matching of character trigrams between query strings and entry titles by using the IR tool Lucence [8], then continued with filtering top hits returned by fuzzy matching. Plenty of systems [37, 55, 1, 80, 15] retrieved a list of candidate entities from the KB by directly retrieving the mention strings with Lucence. The collaborative clustering approaches [12, 21] aimed at finding linked KB nodes for all recognized entities in background documents. Those new knowledge features were then used for assisting the linking of query mentions.

2.2.3. Candidate Ranking

Typical candidate ranking methods are categorized into unsupervised similarity computation, IR and supervised classification methods. Honnibal and Dale [37] ranked the candidates with the cosine similarity and overlapping tokens between their Wikipedia pages and the background documents. Some systems validated the similarity of candidates by using IR approaches [97], or further interpolating of the Bag-of-Word (BOW) similarity [34, 15] and semantic similarity based on Wikipedia concepts [34, 54]. Janya system [90] treated the candidate ranking as entity disambiguation problem and extended the BOW model by introducing profile features and topic model features.

The most popular approaches are learning to rank (L2R), whose goal are to automatically build a ranking model by learning ordinal information from training data. The first study

[7] http://en.wikipedia.org/wiki/IBM
[8] http://lucene.apache.org/

of applying the L2R on EL was reported by Li et al. [55], who adopted a listwise L2R approach – ListNet [11] to model the complete KB candidate orderings for each query. Namee et al. [67], Zhang et al. [123] and Xu et al. [43] trained pairwise ranking SVM [45] classifiers on different classes of entity-level and document-level features. Anastácio et al. [4] discussed the effectiveness of pointwise, pairwise and listwise L2R models for linking and clustering entities.

Supervised methods learn with features on measuring local contextual information for the query mentions. Some systems incorporated global entity information from Wikipedia to make decision on EL. TAC 2010 and 2011 top systems [80, 21] implemented entity disambiguation methods by utilizing category and contextual information from Wikipedia entity pages [20]. On the level of system architecture, some systems aggregate the contribution of multiple collaborative ranking methods [14] or take advantage of combining results from multiple entity linking systems [12] in order to reach the global optimal linking and clustering results.

2.2.4. NIL Detection and Clustering

Besides achieving high answer coverage and linking accuracy, EL systems shall correctly identify "NIL" queries which match no KB entry. NIL detection methods with poor performance can bring significant loss in the final evaluation score. The component with higher NIL accuracy may still lead to lower overall performance if they accomplish higher NIL accuracy by ascertaining too many linked queries as NIL. Many EL systems achieved both higher linking and NIL detection accuracy. The NIL detection is usually involved with two-stage decision making based on results from forerunning components.

- The first stage is fairly simple and straightforward. A query with no results from candidate generation is output as NIL [97]. Some systems [37, 80] sought for candidates in the whole collection of Wikipedia pages, and accordingly a query is set to NIL if its top matched Wikipages are not included in KB.

- The second stage handles NIL detection with supervised classification and ranking models.
  - The NIL detection is integrated into the candidate ranking component by considering NIL as one special KB entry [67, 68].
  - Regarding the confidence score resulting from candidate ranking component, a query is set to NIL if top ranked entry's score is below a pre-defined threshold learned from the training data [80, 15, 122].
  - Some systems [55, 123, 54] constructed individual NIL detection classifiers to predict the absence of entities from KB.

KBP2011 required participants to cluster NIL queries which refer to the same entity, despite it is not included in the KB. Most systems begins with the simple clustering strategies of grouping queries by matching of surface strings, normalized names or coreference mentions from documents [12, 68, 79, 72]. The simple strategies can gain reasonably high performance as only 7.1% of the NIL queries are ambiguous and name variation is easy to resolve. [41].

Some systems [126, 124, 68] used similar feature groups as those of candidate ranking methods for NIL clustering algorithms. HLTCOE 2011 system [68] judged whether pairs of candidates are co-referent by a pairwise classifier, which used the same two-stage approach and identical features as those in their EL system in 2010. Sophisticated methods, including hierarchical agglomerative clustering methods [43, 72], topic models such as Latent Dirichlet Allocation [124] and graphic-based clustering [124, 4] are used to further cluster NIL entities generated by the initial simple clustering. LCC [72] and DAI [78] employed multiple stages of clustering to resolve polysemy and synonym within entity clusters step by step.

Considering the connection between EL and NIL clustering, most systems are featured with *deductive* approaches, i.e., to first link queries with KB and group the rest NIL mentions into clusters. The alternative *inductive* approach [72] is to cluster all queries first with EL

results as features, then further dive and merge preliminary clusters to assign KB ID or NIL. The inductive strategy in LCC system contributes to best performances of monolingual and cross-lingual EL in KBP2011.

2.3. Approaches to Cross Lingual Entity Linking

The CLEL task proposed in KBP 2011 brings new challenges to the traditional EL task. Systems shall introduce approaches to link entities with KB by transferring information across the language barrier. Based on the way of processing cross-lingual queries and KB, CLEL systems in KBP 2011 are broadly classified into two pipelines as demonstrated in Figure 2.1.

- **Pipeline A** (Name Translation and Machine Translation + English EL). Two top systems (CUNY [12] and HLTCOE [68]) submit Chinese query and its associated document to statistical machine translation or dictionary-based translation system to obtain their translation. Then the remaining problem is just the same as in monolingual EL to link translated entities with English KB.
- **Pipeline B** (Chinese EL + Cross-Lingual KB mapping). In contrast with the translation of input queries in **pipeline A**, some top systems (such as LLC [72] and HITS [27]) construct the Chinese counterpart of English KB by mining cross-lingual hyperlinks in Wikipedia, run a Chinese mono-lingual EL system to link Chinese queries to Chinese KB, and finally map the Chinese KB node to English KB node by matching cross-lingual KB linkages.

Both pipelines achieved good performances in the final evaluation of CLEL, although each of them has inherent limitations because of inaccurate translation in **Pipeline A** and insufficient coverage of cross-lingual KB linkages in **Pipeline B**. We will review various representative approaches from CLEL systems in KBP 2011.

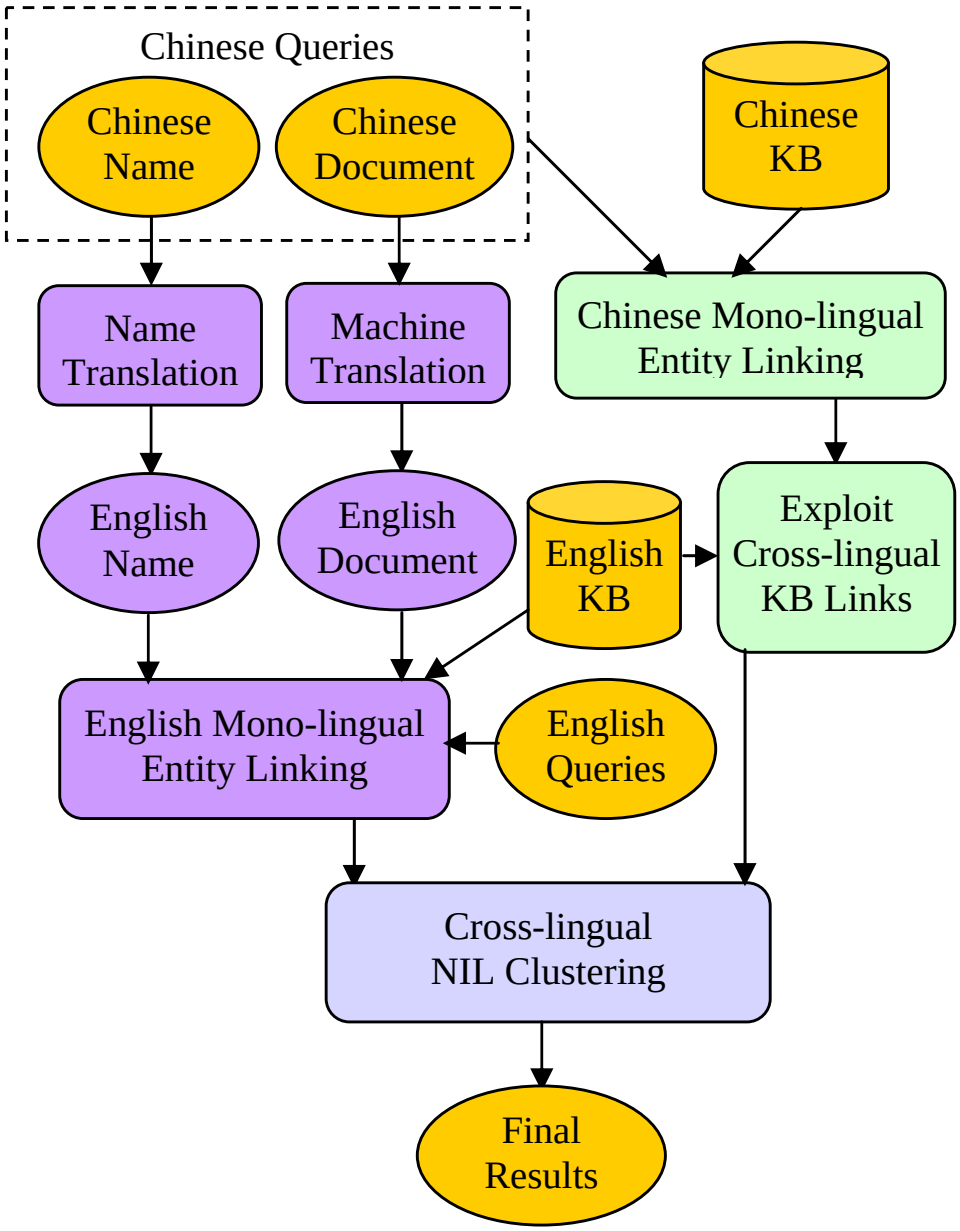

Figure 2.1.. General Cross-Lingual Entity Linking System Architecture [41].

2.3.1. Cross-Document Coreference Approach

Language Compute Corporation (LCC) team [72] converted the EL with NIL clustering into cross-document coreference problem. Their inductive approach of coupling the entity clustering with output from the entity linker has achieved the best performance in KBP 2011 for both mono- and cross-lingual EL.

LCC's mono-lingual systems achieved top results in KBP 2009 and 2010. To make their system language-independent, they construct the Chinese Knowledge Base for the CLEL task by mining Wikipedia cross-language links. As the English KB is derived from English Wikipedia which is closely connected with Chinese Wikipedia, the English Wikipedia serves as the intermediary to align Chinese Wikipedia with English KB. As depicted in Figure 2.2, cross-language links extracted from Wikipedia connect a Chinese entry with its corresponding English entry. Those English entries are connected with entities in TAC KB. The part of Chinese Wikipedia linked with English KB is utilized as the Chinese KB. Both the Chinese and English entity linker are derived from LCC's English EL system based on machine learning and heuristic approaches by combining contextual, surface and semantic features into the ranking score [54].

LCC's cross-document entity coreference system adopts a four-stage clustering algorithm. Each stage of their algorithm deals with different problems.

1. **Surface String-based Initial Clustering.** Entity mentions with identical normalized surface strings are gathered in one subset, each of which consists of mentions in only one language. Both the development and evaluation datasets contain hardly any cross-lingual clusters. [9] For example, all mentions with text "李娜" (Li Na) are grouped in one subset even they refer to different people.

2. **Polysemy Division.** Mentions within one subset are polysemous. To distinguish polysemy in each subset, initially each mention in a subset represents a singleton

[9] All the $1,481$ Chinese entities and 695 entities formed 979 clusters, with only 26 of them being cross-lingual. Only 22 of these cross-lingual clusters link with KB nodes, which accounts for less than 1% cross-lingual clustering.

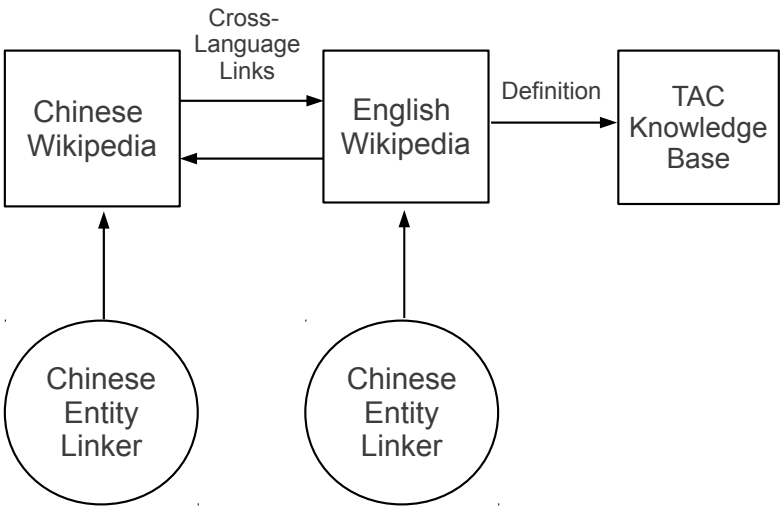

Figure 2.2.. Language-independent Knowledge Base for Entity Linking.

cluster, then the supervised agglomerative clustering algorithm with the standard pairwise model is used to separate mentions with different references. The distance between clusters M_1 and M_2 is the average of relatedness across all pairs of mentions between the two clusters.

$$d(M_1, M_2) = \frac{1}{|M_1| \cdot |M_2|} \sum_{m_1 \in M_1} \sum_{m_2 \in M_2} f(m_1, m_2) \tag{2.7}$$

where the function f estimates the relatedness between m_1 and m_2 with a logistic classifier trained on various pairwise features. The clustering procedure terminates when the current largest value of d is less than a threshold.

3. **Synonymy Merging.** A further issue that can be addressed is that different clusters produced in Step 2 might not correspond to distinct entities. The following merging strategy is subsequently applied to tackle this issue.

   Two clusters are considered as synonymous and merged into one bigger cluster if

the following condition is satisfied.

$$\sum_{m_1 \in M_1} \sum_{m_2 \in M_2} \alpha_k I_k(m_1, m_2) > \kappa,\ k \in (1, 2, \ldots), \tag{2.8}$$

here $\kappa \propto |M_1| \cdot |M_2|$. Two Boolean indication functions I_1 and I_2 are used. I_1 is true if m_1 and m_2 are linked to the same KB or Wikipage with confidence score $>$ 0.75, and I_2 is true if m_1 and m_2 are both embedded in a longer common phrase extracted from their own source documents separately.

4. **KB Link Resolution.** The linkage for a cluster is based on linkage of its containing mentions. Each mention in the cluster is linked to a KB node or NIL using their entity linker, eventually the linking identifier for a cluster is determined by the majority voting of linking identifiers of its containing mentions with random tie-breaking.

Besides the language-independent pipeline, they also performed cross-lingual experiments on translating queries and documents into English by online translation service and processed them with the English EL system, which produced worse results than the Chinese system. They also combined the results from the Chinese and translation systems with an empirical voting strategy. The combining system reached the best performance on the development dataset, nevertheless it performed 0.5% worse than the Chinese system in evaluation.

2.3.2. Deductive One-model-for-all-language Approach

Heidelberg Institute for Theoretical Studies (HITS) system also followed the **Pipeline A** and introduced one general model for all languages. Unlike the *inductive* approach in LCC system, HITS system used the *deductive* approach, i.e. to run entity linker and cluster successively and independently. Their method is similar to the *Text Wikification* task [19], which recognizes the key phrases in a text (*keyword extraction*), and then links these phrases to the most likely Wikipedia concept (*word sense disambiguation*). Different from wikificaiton, HITS system made use of all possible concepts (Wikipedia pages) for

each phrase instead of the most likely concept, and EL system only needed the wikification result of the query mention as the linked entry.

Their system approaches the cross-lingual tasks in three stages:

1. **Context Disambiguation.** HITS system deployed a simplified version of Wikification. The construction of concept begins with *keyword extraction*, which determines whether to identify an N-gram term as the link to Wikipedia concept with the scoring of keyphraseness proposed by Mihalcea and Csomai [19]. They hereby train an SVM classifier with various features derived from Wikipedia linking and contextual information, to identify the most probable Wikipedia concepts for each contextual key terms. The concepts extracted from the context are utilized as a language-independent concept-based representation of a query for identifying wikification of the query mention as the final result.

2. **Entity Disambiguation.** For query mentions with more than one concept candidates, HITS system approaches supervised entity disambiguation using an SVM-based and a graph-based approach. Besides the common context and surface features widely used for candidate ranking, they construct more resources-intensive features, such as relatedness of concept pairs measured by incoming links, outcoming links and category information inherent in Wikipages.

3. **NIL Clustering.** The remaining queries with no linking are clustered by employing a string matching heuristic and spectral clustering. They also deal with cross-lingual clustering by clustering equivalent bilingual queries matching in a dictionary.

The cornerstone of their approach is a *multilingual KB* extracted from multilingual versions of Wikipedia and linked with the TAC KB, which is a subset of English Wikipedia. The multilingual KBs enable the extraction of multilingual Wikipedia to overcome linguistic lexical specificities, therefore even languages with poor resources can take advantage of information in English Wikipedia. To build the multilingual KB, they first use the interlingual links between different language versions of Wikipedia, and extract all the missing links by

enforcing the transitivity properties of Wikipedia pages. They also use external links, image and template information to generate more mappings. At last they apply a supervised filtering method to reduce the noisy mappings.

The advantage of HITS system includes large coverage of multilingual KBs and sophisticated supervised techniques fuelled with resources-intensive features, which are easily adaptable to CLEL for other languages with low-density Wikipedia collections. Their system also achieved the best results in the NTCIR 9 shared task on Cross-lingual Link Discovery [92], whose goal is to link English Wikipedia pages to Korean, Chinese and Japanese target documents.

2.3.3. Transliteration and Translation-based Approach

Contrast with language-independent approaches from **Pipeline A** which maps the KB representation into the query representation space, systems following **Pipeline B** is built on mapping the query representation into the KB representation space.

The query translation often suffers from the problem of translation ambiguity, and this problem is amplified due to the limited amount of context in short queries and various translated counterparts. Most EL queries are involved with ambiguous organization and location names or obscure person names. A poorly-translated query can decrease the effectiveness of EL systems due to bringing in more ambiguity and errors.

To address the problem, the **JHU HLTCOE** [68] system creates multiply English equivalents of Chinese queries by applying bilingual dictionaries and thesauri, interpreting document contexts with statistical machine translation (SMT) methods, and running cross-lingual IR methods, which consider all translations as equivalent queries.

They develop a two-phase Name Transliteration.

1. **Rule-based Translation.** The first phase relies on the bilingual phrase dictionary. To improve the coverage, they use multiple sources of Chinese-English name translit-

erations extracted from newswires and Wikipedia.

2. **Chinese Pinyinization.** 77.2% of all queries from KBP2011 evaluation data can be translated by consulting dictionaries. An orthographic-based transliteration system is trained on Chinese Pinyin (the official system to transcribe Chinese characters into Latin scripts) to deal with the rest queries, since Chinese person and location names are usually presented as Pinyin in English.

The selection of translation also depends on the immediate context around the word to be translated. HLTCOE system uses a hierarchical phrase-based machine translation system [16] to translate Chinese documents into English. They conduct minimum error rate training with the following features for translation decoding.

- SCFG (Synchronous Context-Free Grammar) translation rule score;
- SCFG translation rule arity (number of non-terminals);
- Language model score (with a 3-gram English language model trained on the training data and English Gigaword);
- Word penalty;
- Rule-based translation for numbers.

In the end, they use two strategies to enhance the performance of SMT. They observe a large proportion of untranslated Chinese terms are probably named entities, so they further transliterate those out-of-vocabulary terms.

Using the various translations from dictionaries and SMT systems as queries, the CLEL is treated as a Cross-Language Information Retrieval (CLIR) problem, which retrieves the collection of Wikipages related with the KB entry. They use a structured query translation approach: different translations for a query term are considered to be synonyms. They balance the relative importance of translated terms with their translation probability.

2.3.4. Interlingual Representation Approach

Similarly, the cross-lingual system of **City University of New York** (CUNY) also include name translation and document translation. They expand each query by running their *Chinese name coreference resolution* system on the source document, and then apply different translation approaches on the expanded queries. The SRI hierarchical phrase-based SMT system [127] is trained for translating background documents.

To reduce the influence of translation errors and learn the fine-grained contextual information, they design a novel approach on joint learning the relatedness between entity and KB entry based on *local* monolingual evidence and *global* cross-lingual evidence. They knowledge-based inference is based on following two methods.

Contextual Profile Inference Network. To support entity identification and disambiguation, monolingual evidence is extracted and estimated by calculating co-occurrences between the query mention and its profiles from a background document. Intuitively "profile" of an entity, such a person's title, origin and employer can help identifying the entity. For example, three different entities with the same name "阿尔伯特/Albert" can be distinguished by their respective context entities(*profile*): "比利时/Belgium", "国际奥委会/International Olympic Committee" and "美国科学院/National Academy of Sciences".

To leverage the local contextual information, they first run their slot filling system to extract related profiles for a entity mention and derive the relatedness between profile and entity using *Information Networks* [56]. This method is good at dealing with entities with common organization or person names.

Cross-lingual Supporting Matrix. If no sufficient profiles for an entity are found in local context, its profile entities are processed by the slot filling system and given with slot attributes, which are also used for reinforcing the relation between the entity and KB. Hence, **interlingual representation** is proposed to recover the hidden alignment between the entity with its comparable and related KB in another language. CUNY system adopt a holistic approach [118] on exploiting both transliteration similarity and monolingual co-

occurrences for mapping entities and KB entries into a common interlingual representation.

The co-occurrence between each Chinese entity mention and each English KB entry is stored in a large entity supporting matrix. Its calculation consists of three steps:

1. **Initialization:** computing basic cross-lingual similarities R_{ij} between a name mention i and each corresponding KB node j using transliteration similarity score described in [118], topic modeling results and the similarity between the document and the KB entry article.

2. **Reinforcement model:** iteratively reinforcing the translation similarities R_{ij} by exploiting the monolingual co-occurrences with respective neighbours.

   We denote a neighbour of a term t in the supporting matrix as $\mathcal{N}(t)$. The procedure of finding neighbours is depicted in Figure 2.3.

   The neighbour of a name mention in a source document are defined as a profile entity generated from the slot filling system or an entity associated or concurrent with the name mention after coreference resolution. For example, given the query q: "**Sevilla**" in Figure 2.3, the entity "**Spain**" is determined as $\mathcal{N}(q)$ due to its co-occurrence with "**Sevilla**".

   The neighbour of a KB entry are entries which link or are linked to the KB entry in corresponding Wikipedia pages. For "*Seville*" in Figure 2.3, we go to the original Wikipage, where all the linked pages are selected as $\mathcal{N}(node)$, such as "`Spain`" and "`province of Seville`", whose corresponding cells in $\{R_{ij}\}$ are "*Spain*" and "*Seville (province)*" respectively. Additionally Wikipages with links to "`Seville`" are also considered as $\mathcal{N}(node)$.

   The KB entry candidates for a query mention are generated from the **Candidate Generation** component. KB candidates for "**Sevilla**" includes "`Seville`" , "`Seville (province)`", "`Seville, FL`" and etc..

   The iterative reinforcement model [118] is propagated as follows. Let R_{ij}^t denote the

similarity of a name mention node i and a KB j at t-th iteration:

$$R_{ij}^{t+1} = \lambda \cdot \left[\sum_{(u,v)_k \in B^t(i,j,\theta)} \frac{R_{uv}^t}{2^k} \right] + (1 - \lambda) R_{ij}^0 \tag{2.9}$$

The model is expressed as a linear combination of the relational similarity $\sum \frac{R_{uv}^t}{2^k}$ and transliteration similarity R_{ij}^0. $B^t(i,j,\theta)$ is a set of best matched pairs of the form ([*name mention*, *documentID*], *KB entry*) selected from neighbours with criterion:

$$\forall (u,v)_k \in B^t(i,j,\theta), R_{uv}^t \geq \theta, \tag{2.10}$$

where $(u,v)_k$ is the pair with k-th highest similarity score. θ is a predefined threshold.

3. **Entity Extraction.** For a query mention denoted as row i in the matrix, the KB node with the maximum score of the row is chosen as the target entry. The propagated matrix is also applicable for other related tasks such as entity clustering and name translation mining.

The advantage of interlingual representation is that it does not require machine translation and native KB construction. The holistic approach is flexible to learn from transliteration dictionaries, parallel or comparable corpora, such as Wikipedia. It is thus applicable for CLEL for all languages.

2.4. Conclusion

The concept of mining knowledge from large-scale KB was investigated quite intensively in recent years. TAC conference continues proposing series of monolingual and cross-lingual KBP tasks and providing resources and infrastructures for those large-scale NLP tasks. In this chapter, we thoroughly reviewed the different system architectures and underlying components employed in the past EL and CLEL tasks. In the following chapters we will

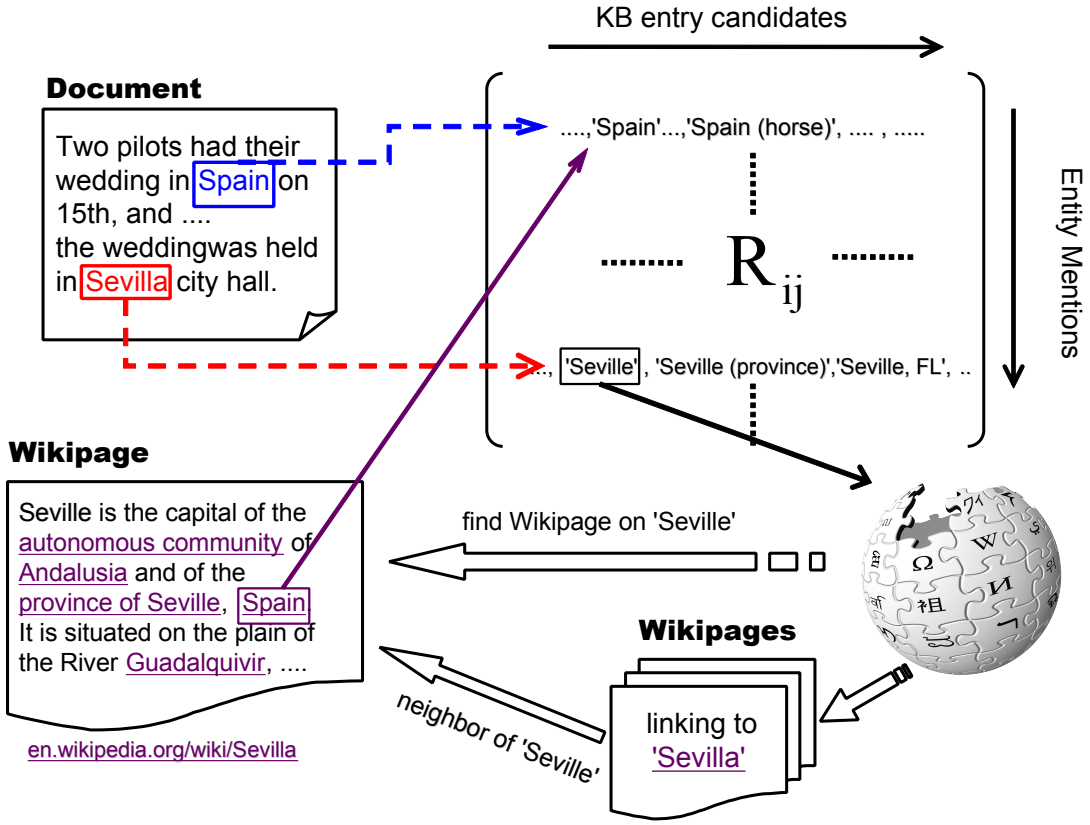

Figure 2.3.. Example of Neighbour nodes for entity mentions and KB nodes. First we find contextual neighbours of a query mention in background document. We generate KB entry candidates for the query mention and neighbours, which formalize the supporting matrix $\{R_{ij}\}$. The KB Node candidates are displayed in the row for q. Then we build neighbour relations between KB candidates by referring to anchor linkings in Wikipedia. For example, we find the Wikipage on "`Sevilla`" linking to Wikipage "`Spain`", therefore "`Spain`" is a KB neighbour of "`Sevilla`". The Wikipage on "`Andalusia`" links to "`Sevilla`", so it is also a neighbour of "`Sevilla`"

introduce our systems in the CLEL of KBP 2011. Chapter 3 presents the resources and tools used for the implementation of our system, and then Chapter 4 focuses on system description, evaluation results and new experiments for Chinese to English EL.

3. Experimental Methodology

This chapter will introduce a variety of resources used for the implementation and training of our cross lingual Entity Linking (CLEL) system, including Wikipedia Knowledge Base (KB) and document resources, as well as various tools for Natural Language Processing (NLP), Information Retrieval (IR), and Wikipedia dump processing. Subsequent chapters detailing experiments will refer to the descriptions here. Section 3.1 introduces the linking structure of Wikipedia. Section 3.2 describes the English KB provided by TAC and the construction of Chinese KB based on Chinese Wikipedia. Sections 3.3 and 3.4 summarize the NLP tools for English and Chinese separately.

3.1. Wikipedia Linking Structure

Wikipedia is a multilingual, collaborative encyclopedia on the web which is freely available for research purposes. On December 31, 2011, there were over 3.8 million articles in English [1] and versions available in 283 languages [2]. Wikimeida Foundation [3] provides periodical snapshots of the up-to-date Wikipedia dumps in the form of wikitext sources and metedata embedded in XML.

Wikipedia is structured as an interconnected network of articles. Each article in Wikipedia is involved with an unique title as its identifier. For example, the article [4] for the country of

[1] http://www.wikistatistics.net/wiki/en/articles/full
[2] For the complete statistics refer to http://meta.wikimedia.org/wiki/List_of_Wikipedias
[3] http://download.wikimedia.org
[4] http://en.wikipedia.org/wiki/Germany

Germany has the canonical URL suffix "*Germany*". Each article is enriched by explanatory hyperlinks (namely *wiki links*) and tags, which are named according to consistent patterns and meaningful interpretations. The built-in wiki links point to other articles and a list of corresponding pages in other languages. We take advantage of the links and pages extracted from Wikipedia for approaching EL.

A typical text in the source of Wikipedia article on "Germany" looks like [5]:

> "Germany, officially the Republic of Germany is a [[*federation*|federal]] [[*parliamentary republic*]] in [[*Europe*]]. The country consists of [[*Lands of Germany*|16 states]] while the [[*capital city*|capital]] and [*[List of cities in Germany by population*|largest city]] is [[*Berlin*]]."

Wiki links are signified by the pair of double brackets, and automatically connected to other Wikipedia entries. This snippet holds three article links to other Wikipedia pages, titled "parliamentary republic", "Europe" and"Berlin". Other links are divided by the vertical bar into two parts: the first is the title of a linked article, e.g. "Lands of Germany", while the succeeding is the anchor text displayed on current Wikipedia page, e.g. "16 states".

Category links point to a special "Category" page, which consists of articles a related topic defined by the category name. For example, [[Category:Germany]] and [[Category:Alpine countries]], which directly link to category pages (with titles starting with *Category:*). We determine the categories of the Wikipedia pages by those links.

The Wikipedia page in one language is often linked to a multilingual database of corresponding terms by **interlanguage links**. For example, English article "Germany" contains interlanguage links including [[de:Deutschland]] connecting with German Wikipedia [6] and [[zh:德国]] with Chinese one [7]. They link to the comparable German article on "Deutschland" and the Chinese article on "德国". In a similar manner Germany and Chinese articles

[5] Bold format an other metadata are omitted for the clarity of the example.
[6] http://de.wikipedia.org
[7] http://zh.wikipedia.org

contain the interlanguage links to English counterparts.

A **redirect page** provides alternative titles for a target page [8]. It provides spelling variations and grammatical variants, such as English article titled “Deutschland” redirects to article on “Germany”. Redirect pages also represent acronyms or alternative names, such as “USA” for the article “‘United States”, and the historical name “北平”(Beiping) for “北京”(Beijing).

A **disambiguation page** resolves the ambiguity of terms. It serves as the inventory of wiki links pointing to the correct article titles. It contains brief explanatory content for each article. For instance, the page “Germany_(disambiguation)” [9] contains 21 links, including a political entity “German Empire”, a baseball player “Germany Schaefer” and a landmark “Germany Valley”.

3.2. Knowledge Resources

For the CLEL task, we use two language-dependent KBs — official TAC KB and a Chinese KB which we extracted from the Chinese Wikipedia, as well as knowledge repositories extracted from the Wikipedia linkage structure.

3.2.1. TAC Knowledge Base

The U.S. National Institute of Standards and Technology (NIST) held comparative evaluations for English EL systems from 2009 to 2011 and introduced CLEL systems in 2011 as a part of Knowledge Base Population (KBP) tracks [10]. See Section 2.1.1 for a detailed description of various tasks.

The English KB was derived from Wikipedia snapshot cached in October, 2008 and kept used for KBP tracks since 2009. The English KB consists of 818,741 entries. Each

[8] http://en.wikipedia.org/wiki/Wikipedia:Redirect
[9] http://en.wikipedia.org/wiki/Germany_(disambiguation)
[10] See www.nist.gov/tac/

entry has a entry *ID* (canonical identifier) and a *title* of the Wikipage, an *entity type*, an automatically parsed version of data from the infobox in the page, and a stripped version of the Wikipage. Some articles were discarded because of parsing errors resulting from Wikipedia markups, therefore a subset of Wikipedia article collection was assembled as the TAC KB. A sample entry from KB is depicted in Figure 3.1, while Figure 3.2 shows its Wikipedia source.

```
<entity wiki_title="Theodore_Roberts" type="PER" id="E0000003" name="Theodore
    Roberts">
<facts class="infobox actor">
  <fact name="birthdate">October 8, 1861 (1861-10-08)</fact>
  <fact name="birthplace">San Francisco,California, U.S.</fact>
  <fact name="deathdate">December 14, 1928 (aged 67)</fact>
  <fact name="deathplace">Hollywood, California</fact>
  <fact name="restingplace">Hollywood Forever Pineland 124</fact>
  <fact name="occupation">Film, stage, actor</fact>
</facts>

<wiki_text><![CDATA[Theodore Roberts
Theodore Roberts the actor is not to be confused with author Theodore
    Goodridge Roberts, 1877-1953, who wrote the "The Harbor Master". Please
    see.
Theodore Roberts (October 8, 1861, San Francisco, California -- December 14,
    1928, Hollywood, California) was an American movie and stage actor. He was
    a stage actor decades before becoming lovable old man in silents. On stage
    in the 1890s he acted with Fanny Davenport in her play called Gismonda
    (1894) and later in The Bird of Paradise (1912) with actress Laurette
    Taylor.
He started his film career in the 1910s in Hollywood, and was often associated
    in the productions of Cecil B. DeMille.

Selected filmography
]]></wiki_text>
</entity>
```

Figure 3.1.. Sample Knowledge Base Entry on "*Theodore Roberts*". The `wiki_title` "**Theodore_Roberts**" is the base name of URL of the Wikipedia page. The `type` of entry is **PER** (person). The KB `id` is **E0000003**. The `facts` class covers content from the Wikipedia infobox. The `wiki_text` is the stripped version of Wikipedia article.

Based on the type of infobox in the original article, each entity in KB was automatically assigned with one of four types: PER (person), ORG (organization), GPE (geo-political) entity or UKN(unknown), such as the *Infobox Actor* belongs to the NE type of person. After automatically mapping infoboxes to entity types, The KB consists of 116,498 GPE, 55,813 ORG, 114,523 PER and 531,907 miscellaneous/unknown entities.

```
<page>
    <title>Theodore_Roberts</title>
    <id>6831454</id>
<text xml:space="preserve">
''Theodore Roberts the actor is not to be confused with author [[Theodore
   Goodridge Roberts]], 1877--1953, who wrote "The Harbor Master".  Please see
    [[Talk:Theodore Roberts|discussion page]]..''
{{No footnotes|article|date=February 2008}}
{{infobox person
|image=TheodoreRoberts.jpg
|birth_date={{birth date|1861|10|8}}
|birth_place=[[San Francisco, California]],<br> [[United States]]
|death_date={{death date and age|1928|12|14|1861|10|8}}
|death_place=[[Hollywood, California]],<br> [[United States]]
|restingplace=Hollywood Forever<br>Pineland 124
|occupation=[[Film]], [[Theatre|stage]] [[actor]]
}}

'''Theodore Roberts''' (October 8, 1861, [[San Francisco]], [[California]] &
   ndash; December 14, 1928, [[Hollywood]], [[California]]) was an American [[
   film actor|movie]] and [[stage actor]]. He was a stage actor decades before
    becoming lovable old man in silents. On stage in the 1890s he acted with
   [[Fanny Davenport]] in her play called ''Gismonda'' (1894) and later in ''
   The Bird of Paradise'' (1912) with actress [[Laurette Taylor]]. He started
   his film career in the 1910s in [[Hollywood]], and was often associated in
   the productions of [[Cecil B. DeMille]]. He was buried in [[Hollywood
   Forever Cemetery]].

==Selected filmography==
{|class=wikitable
!Year!!Title
|-
|rowspan=4|1914||''[[The Ghost Breaker]]''
...
...
[[Category:1861 births]]
[[Category:1928 deaths]]
[[Category:American stage actors]]
[[Category:American film actors]]
[[Category:American silent film actors]]

{{US-film-actor-1860s-stub}}
{{US-theat-actor-1860s-stub}}

[[de:Theodore Roberts]]
[[fr:Theodore Roberts]]
[[sv:Theodore Roberts]]

</text>
</page>
```

Figure 3.2.. Sample Wikipedia Source text of the page on "*Theodore Roberts*" included in the English Wikipedia dump.

3.2.2. Collections of Source Documents

Each query entry for EL takes the form of [name-string, docid]. The query with **name-string** occurs in a background document with **docid**. The background document provides associated contexts which may be useful for linking the query with KB.

KBP 2009 provided a huge collection of source data, from which background documents were chosen. The corpus consisted of documents covering ACE 2008 evaluation [11] source data and newswire texts from English Gigaword Fourth Edition [12]. KBP 2010 introduced new web document collection comprising of web pages to test how EL systems perform on noisy and unstructured web data. KBP CLEL 2011 used documents from Chinese Gigaword Fourth Version [13]. The statistics of source collections are listed in Table 3.1.

Genre	#documents
Broadcast Conversation	17
Broadcast News	665
Conversational Telephone Speech	1
Newswire	1,286,609
Web Text	490,59
Chinese Gigaword	∼1M

Table 3.1.. Number of documents in source collections.

3.2.3. Chinese Knowledge Base

The CLEL task is brought out by KBP 2012 to connect a Chinese entity with corresponding English KB entry. In addition to the challenges in monolingual EL, The key problem in CLEL is to break through the language barrier. We address this problem of mapping the KB representation into the query representation by creating Chinese KB.

The construction steps are detailed as following:

[11] http://www.itl.nist.gov/iad/mig/tests/ace/2008/
[12] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2011T13
[13] http://www.ldc.upenn.edu/Catalog/catalogEntry.jsp?catalogId=LDC2009T27

Cross lingual Mapping Extraction. We use the interlanguage links to create a Chinese to English title mapping. Since the amount of Wikipages is expanding every day, we choose the latest English and Chinese Wikipedia for maximum linkages between bilingual pages. English dump is created on 26 May, 2011 and Chinese dump on 21 May, 2011. They are archived almost simultaneously, so interlanguage links shall have high consistency.

The proportion of Chinese entries with English links is larger than that of English entries with Chinese links. This is probably due to the fact that English Wikipedia grows at faster rates than other versions. We extract bi-directional cross-lingual title mappings, the union of which is employed as the cross-lingual mapping dictionary.

Connecting English Wikipedia with English KB. The edition of English Wikipedia in 2011 contains approximately 3.8 million articles, almost five times more than those in 2008 edition. We create mappings between articles of those two editions automatically by matching article IDs and titles, as well as titles of redirect and disambiguation pages from the 2011 edition.

Some article titles are modified due to the evolution and proliferation of structural content in Wikipedia.

A regular article is changed into a redirect page, in this case we rebuild the mapping between redirect title and the new page.

A regular page is reorganized as a disambiguate page. The original one is given a title with a specific constraint and absorbed in the disambiguate page. For example, the page titled “Wang Li” in 2008 Wikipedia is renamed as “Wang Li (linguist)” [14] in 2011. The new version on “Wang Li” [15] is defined as a disambiguation page with “Wang Li (linguist)” as a listed disambiguating link.

To deal with the name shifting, we compare the cosine similarity between a KB article and all reference pages in the disambiguation page. The most relevant page

[14] http://en.wikipedia.org/wiki/Wang_Li_(linguist)
[15] http://en.wikipedia.org/wiki/Wang_Li

is chosen as the mapping to the KB. For example, “Wang Li (linguist)” in Wikipedia 2011 connects to KB entry titled “Wang Li”.

The usage of latest Wikipedia edition provides more entities and contents than the old version. The linking information is beneficial for building entity linking and clustering components in Chapter 4.

3.2.4. Wikipedia Processing Tools

The tools described in this section process the English and Chinese Wikipedia dumps. Each Wikipedia page is wrapped up in the predefined XML format [16], demonstrated in the Figure 3.2.

Comparisons of KB entry in Figure 3.1 and Wikipedia source text in Figure 3.2 indicate that all wiki links and formatting markups are removed from texts in English KB. The cached Wikipages contain some noisy or encoded characters. The goal of Wikipedia processing tools is to remove noisy characters and render decoding characters, such as `&ndash` needs to be decoded to show the correct character, namely “–” for `&ndash`, and extract terms within various Wiki markups and links from the source. The Wikipedia syntactic and linking information is useful for EL.

We modify the tool WikiExtractor [17] to clean the syntactic decorations and html tags and generate only the textual information from Wikipedia dump, including the *ID*, *title*, *plain text*, *Wikipedia links*, *interlanguage links*, and *category links*, and dispose of any other information, such as images, tables, references and lists. Additional databases are created for matching redirect titles with target page titles, one-to-many matching of disambiguation titles to their containing page titles. All Wikipedia pages are saved in the SGML format and indexed with Indri [18] to easily access and efficiently retrieve.

[16] http://meta.wikimedia.org/wiki/Special:Export/Main_Page
[17] http://medialab.di.unipi.it/wiki/Wikipedia_Extractor
[18] http://lemurproject.org/indri/

3.3. English Language Processing Tools

We introduce tools for preprocessing both the English KB and background documents for the CLEL experiments in this section.

Stanford CoreNLP provides a suit of English NLP tools. It analyses raw English language text and produces the lemma of words, their Parts of Speech(POS), named entity (NE) tags, and generates the syntactic parsing trees of sentences and resolves name phrase co-references. We mainly use the POS tagger and NE recognizer, whose results are displayed in Table 3.2.

Sentence:	**Hong Kong's medal ambition at the 2008 Beijing Olympic Games lies in several marginal events.**
POS tagged:	**Hong**/NNP **Kong**/NNP **'s**/POS **medal**/NN **ambition**/NN **at**/IN **the**/DT **2008**/CD **Beijing**/NNP **Olympic**/NNP **Games**/NNPS **lies**/VBZ **in**/IN **several**/JJ **marginal**/JJ **events**/NNS **.**/.
NER tagged:	**Hong**/B-LOCATION **Kong**/I-LOCATION **'s**/O **medal**/O **ambition**/O **at**/O **the**/O **2008**/B-DATE **Beijing**/B-LOCATION **Olympic**/B-MISC **Games**/I-MISC **lies**/O **in**/O **several**/O **marginal**/O **events**/O **.**

Table 3.2.. Example of POS- and NER-tagged Sentence from Stanford CoreNLP.

The Stanford POS tagger based on a Maximum Entropy model [94] is a highly effective tool for English POS tagging. Each token in a sentence is classified as to which POS it belongs , by inferring in a bidirectional dependency network [35]. The model learns lexical features of the previous and current tokens, and effectively deals with unknown word features. The tagger achieves 97.24% tokenization accuracy on the Penn Treebank WSJ data.

The Stanford NE recognizer [29] is a widely-used tool for extracting NEs. It implements a general linear chain Conditional Random Fields [50] sequence models learning with well-engineered features. It provides a robust model trained on a collection of various NE-annotated corpora, which makes the NE recognizer fairly adaptable to multiple domains.

The Stanford NE recognizer can identify the following types: time, location, person, organization, money, percent and date. As shown in Finkel et al. [29], those NEs of location, person and organization domain the training data and can be identified with high

accuracy. The three NE types corresponds to KB types occurring in the TAC KB collection, therefore we consider NE models trained for identifying them.

3.4. Chinese-Language NLP Tools

Chinese texts are written in a sequence of Chinese characters (ideograms). In contrast with English, Chinese characters are written successively without spaces to delimit words. We need to perform *word segmentation* to prior linguistic processing. Given a sentence:

意大利足坛劲旅拉齐奥队12日在里斯本举行一场国际俱乐部友谊赛。

(*Translation:* On the 12th, the strong Italian soccer contingent Lazio held an international club friendly in Lisbon.)

We need to identify the following words in this sentence: 意大利 (Italian), 足坛(soccer), 劲旅 (strong contingent), 拉齐奥队 (Lazio team), 12日 (12th), 在 (at) 里斯本(Lisbon), 举行 (held), 一 (one), 场 (*classifier* [19]), 国际 (international), 俱乐部 (club), 友谊赛 (friendly match).

Chinese word segmentation shall determine the correct sequence of words for a sentence. After segmentation, the preceding sentence is separated as follows:

意大利 足坛 劲旅 拉齐奥队 12日 在 里斯本 举行 一 场 国际
俱乐部 友谊赛 。

Once we break Chinese texts into a sequence of words, Chinese information retrieval or extraction can follow the langauge-independent approaches for English.

Second, POS tags are identified for the sequence of Chinese words. Each word in the sentence will be given a POS tag as defined in Table 3.3.

[19] The classifier "场" measures sport games. For the definition of a Chinese classifier, refer to http://en.wikipedia.org/wiki/Chinese_classifier

意大利/ns 足坛/n 劲旅/n 拉齐奥队/nt 12日/t 在/p 里斯本/ns 举行/v 一/m 场/q 国际/n 俱乐部/n 友谊赛/n 。/ w

There exist many possible ways of textual segmentations. For example, “拉齐奥队” can be broken into two words “拉齐奥”(Lazio) and “队”(team). Contrarily, some tokenizer matches longest words, such as taking “国际俱乐部” (international club) as one word. The Chinese tokenizer handles with out-of-vocabulary and ambiguous word boundary problems. All segmentation methods yield mistakes sometimes. In Chapter 4, we will introduce our methods of dealing with those problems.

Most KB entities are about specific NEs. Contextual NEs in background documents could provide additional information for one entity. The final procedure is to recognize NEs within the sentence. i.e.,

意大利/LOC 足坛/n 劲旅/n 拉齐奥队/ORG 12日/TIM 在/p 里斯本/LOC 举行/v 一/NUM 场/q 国际/n 俱乐部/n 友谊赛/n 。/w

After running the Chinese NE tagger, 意大利 (Italian) and 里斯本 (Lisbon) is identified as a location (LOC), 拉齐奥队 (Lazio team) is recognized an organization (ORG). “NUM” standards for a number term 一 (one). “TIM” standards for a temporal term.

Chinese word segmentation, POS classification and NE recognition have been studied extensively for several decades. We choose one sophisticated and versatile tool [112], developed by the National Laboratory of Pattern Recognition (NLPR), to deal with all those Chinese processing problems. The NLPR tool first executes word segmentation and POS tagging, and then recognizes NE based on the preliminary results. It enhances the hybrid Chinese NE recognition model with heuristic human knowledge, particle features and sub-classes for transliterated person names.

Tag	Description	Tag	Description	Tag	Description
a,ad	Adjective	ns	Place noun	m	Measure
an	Noun adjective	nt	Affiliation	vn	Noun verb
ag	Adjectival morpheme	nx	Non-Chinese character	n	Noun
b	Distinguishing word	nz	Other special noun	w	Punctuation marks
c	Conjunction	o	onomatopoeia	ng	Noun morpheme
d	Adverb	p	Preposition	x	Non-morpheme
dg	Adverbial morpheme	q	Classifier	k	Post-adjective of degree
e	Interjection	r	Pronoun	nr	Proper noun
f	Location	s	Locational noun	y	Modal particle
g	Morpheme	t	Time noun	z	Stative modifier
h	Pre-adjective of degree	tg	Time morpheme	vd	Adverbial verb
i	Phrase	u	Auxiliary	l	Idiom
j	Abbreviation	v	Verb	vg	Verbal morpheme

Table 3.3.. the tagset proposed by Institute of Computational Linguistics at Peking University .

4. Cross-Lingual Entity Linking System

4.1. Introduction

In this chapter we describe our participating system for the Cross-Lingual Entity Linking (CLEL) task in TAC KBP 2011. Given an entity and a background document mentioning it, the entity linking (EL) task is to find whether the entity exists within the knowledge base (KB), or shall be set as NIL otherwise. TAC 2011 proposed a new CLEL task. The KB is a subset of English Wikipedia while the background documents are in either Chinese or English. The cross-language scenario raises more challenges than the previous monolingual task. The main problems of CLEL include:

- mining synonyms of query mentions in documents, i.e. different query mentions can refer to the same entities.
- disambiguating the polysemy of entities, i.e. identical query mentions can refer to different entities.
- connecting knowledge between different languages.

We introduce several underlying components in our system that address these problems.

Our participating system can be categorized as the **Pipeline B** architecture described in Section 2.3, including constructing a Chinese KB and Chinese EL system. The core components of our system include document retrieval and entity clustering. The retrieval module returns the most likely KB entry as the linked target, afterwards clustering algorithms are used to group NIL entities pointing to identical entities.

For comparisons, we also built a pipeline based on Cross-Language Information Retrieval (CLIR). We collected translation candidates from manually constructed dictionaries, phrase tables generated by machine translation (MT) methods, and translation results of an online service. We retrieved Chinese KB with those translated queries and finally combined those retrieved results. However, the CLIR performed worse than the monolingual retrieval of Chinese queries on Chinese KB. Therefore our final system adopts two parallel monolingual pipelines for Chinese and English entities respectively. The framework of the monolingual pipeline is presented in Figure 4.1.

The English and Chinese systems share the same workflow:

Query processing and expansion. NLP tools transforms unstructured queries into the structured format, and enrich their information with query expansion from background documents.

Wikipedia retrieval. IR techniques are applied to retrieve relevant articles with expanded queries from the complete collection of Wikipedia pages. **Interpolation** is used for incorporating retrieved results from different representation of queries, which will be discussed in Section 4.3.

NIL entity Determination. **KB mapping** determines linked entities or NIL by judging whether the title of the most relevant Wikipedia article is mapped to the title of a KB entity [1];

NIL entity clustering. We cluster NIL entities with context feature vectors calculated based on large amounts of relevant documents in Section 4.4.

The fusion of NIL clustering and linking results is taken as the final submission for the CLEL systems.

Our main efforts after the TAC evaluation are made to increase the EL performance over our CLEL systems. We propose a novel CLEL system in Section 4.7 with new entity generation method and a simple generative EL model, which increases the accuracy of

[1] We construct two dictionaries mapping Chinese and English Wikipedia titles to KB entities respectively.

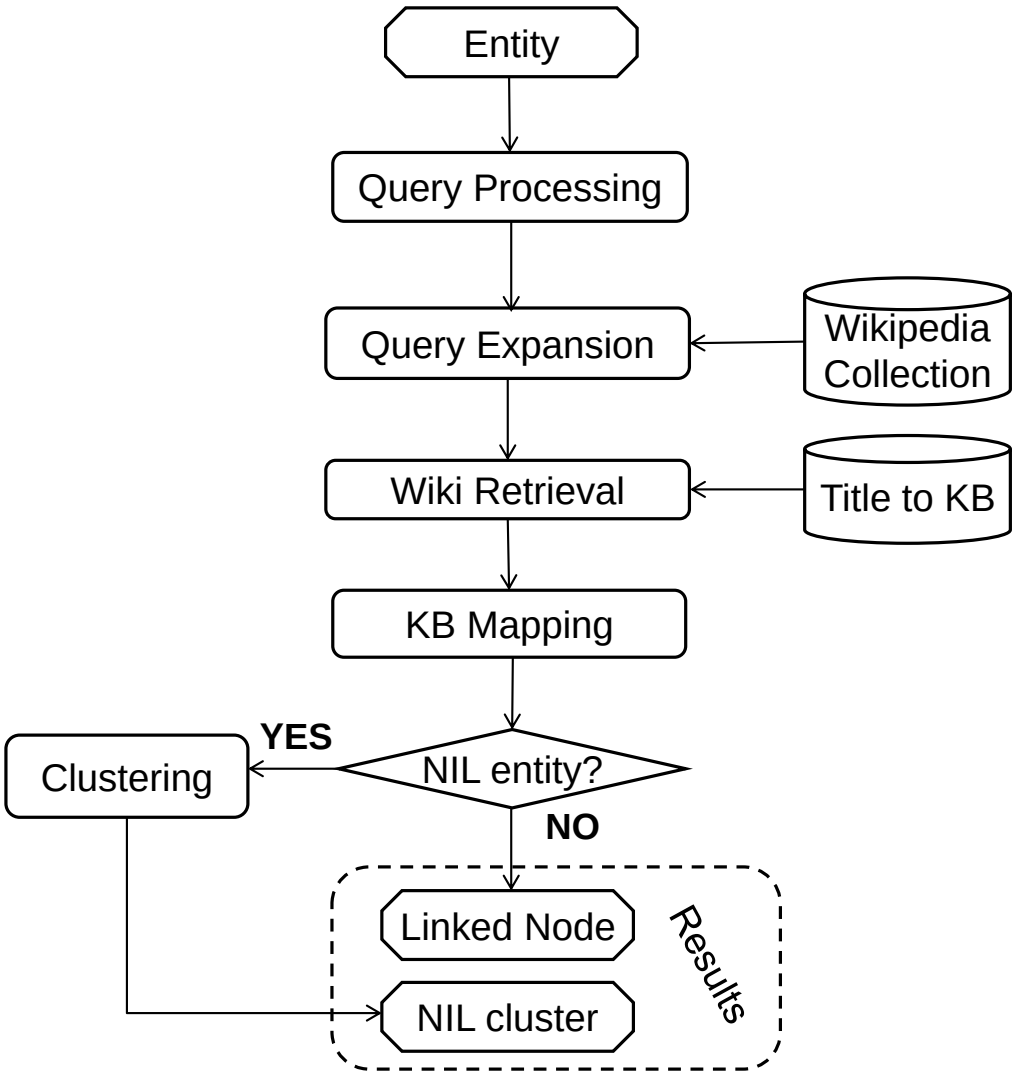

Figure 4.1.. The overall architecture of our Monolingual Entity Linking System.

CLEL significantly.

The rest of the article is organized as follows. We introduce the main components of our system in Sections 4.2, 4.3 and 4.4. Section 4.5 presents the experimental comparisons of cross-lingual and monolingual retrieval results for Chinese entities. Performances of the overall system and individual components are evaluated in Section 4.6. Section 4.7 details the new system. Section 4.8 concludes the chapter.

4.2. Prepossessing

4.2.1. Background Knowledge Extraction

We introduce the bilingual KBs and cross-lingual title mapping in Section 3.2. All those resources serve as knowledge repositories for various components in our system. As shown in Figure 4.1, the collection of retrieved documents consists of all articles extracted from Wikipedia for English and Chinese respectively. The KB mapping is used to determine NIL entities. The WikiExtractor tool introduced in Section 3.2.4 is applied to extract plain articles from both Wikipedia dumps.

Chinese articles are written in traditional and simplified Chinese. Each traditional character is mapped to a simplified character while one simplified character is mapped to several traditional characters, such as the simplified character “尝” to “嘗” and “嚐”.

The CLEL task only provides source documents and queries in simplified Chinese, accordingly we use the mediawiki-based tool[2] to convert traditional Chinese Wikipedia pages into simplified Chinese counterparts.

[2] github.com/tszming/mediawiki-zhconverter

4.2.2. Document and Query Processing

Initially both source documents and Wikipedia articles are preprocessed. The English processing is the same as our previous Slot Filling system [17]. The Chinese preprocessing includes the following steps:

1. Converting html escape characters and removing noisy html garbage (especially for web documents).
2. Segmenting sentences in each document.
3. Breaking a Chinese sentence into sequence of words or Named Entities (NEs).

We use the NLPR tool [112] described in Section 3.4 for segmenting Chinese sentences.

Unlike English, Chinese sentences are written without spaces to delimit words, therefore we need to break each sentence into successive separate tokens. We use the following segmentation methods to define index units for the retrieval component.

1. Breaking a sentence into n-grams — uni-character (individual Chinese character) and bi-character (two consecutive characters);
2. Segmenting a sentence into words by using a word segmentation tool.
3. Recognizing NEs in a word sequence, and generating a mixed sequence of words and NEs.

In Section 4.3 we will combine retrieval models based on those various segmented Chinese texts to relieve the influence of segmentation ambiguities, tokenizing errors and out-of-vocabulary issues.

4.2.3. Acronym Expansion

In order to boost the performance of retrieval module, we adopt a simple method to expand acronym queries. English queries containing all capital letters are considered as acronyms.

Chinese tagging tool contains Chinese vocabulary for acronyms, and therefore is capable of recognizing Chinese acronym words, such as “皖” is short for “安徽省” (Anhui Province), and “人大” for “人民代表大会” (People’s Congress) or “中国人民大学” (Renmin University of China).

Wikipedia linkage information including titles from redirect and disambiguation pages is used to resolve those acronyms. Searching “皖” in Chinese Wikipedia automatically redirect to the page titled “安徽省”. The disambiguation page on “人大” [3] consists of links to articles on both “人民代表大会” and “中国人民大学”. On the other hand we extract the expansion candidates from local contexts in background documents using the following heuristic.

1. If the acronym appears within parentheses, the previous N contiguous tokens are chosen as candidates (N is the number of English letters or Chinese characters in an acronym), otherwise we consider all recognized NEs.

2. The candidates whose initials are identical to the letters from the acronym are chosen as the expansion. If several candidates still exist, we select the one with the largest term frequency in the document. For example, in the text ”*..referring to the National Food Authority (NFA),..*” it is obvious to extract ”*National Food Authority*” as the expansion of the query ”*NFA*”.

4.3. Document Retrieval

We first retrieve the relevant entries from KB for a query. The Chinese tokens for the expansion are extracted from the POS-segmented documents. We retrieve documents from unigram-, bigram-, POS- and NE-segmented corpora.

Table 4.1 shows the segmentation sample and analyses the problem of ambiguous segmentation boundary. The phrase “南京市长江大桥” is ambiguous to segment. It can

[3] http://zh.wikipedia.org/wiki/%E4%BA%BA%E5%A4%A7

Sentence:	南京市长江大桥位于江苏省
Uni-character:	南 京 市 长 江 大 桥 位 于 江 苏 省
Bi-character:	南京 京市 市长 长江 江大 大桥 桥位 位于 于江 江苏 苏省
POS tagged:	南京市/ns 长江/ns 大桥/n 位于/v 江苏省/ns
NER tagged:	南京市/LOC 长江大桥/LOC 位于/v 江苏省/LOC

Table 4.1.. Tokenized representations of Chinese text resulting from different segmentation strategies.

be treated as "南京市 长江 大桥" (*Nanjing Yangtze river bridge*), or alternatively '南京 市长 江大桥" (*Nanjing mayor Jiang DaQiao*), as the character "江" is a popular Chinese surname. The multiple possible segmentations are detrimental to pinpointing retrieved results. The segmentation ambiguity influences the POS- and NE-tagged sequence, while the unigram and bigram character sequences supplement lexical information, such as "长江" (*Yangtze River*) and "市长"(*Mayor*)in the bi-character sequence, which acts as smoothing for word-based retrieval scores. The combination of indexing via different representations of character sequences is more robust than each individual indexing. Simple linear interpolation of retrieve scores generate better ranking orders.

To deal with segmentation ambiguities, tokenizing errors and out-of-vocabulary issues, we first retrieve indices built with different segmentations of queries and documents, combine those retrieval scores with linear interpolation, and finally rerank relevant KB entries according to the combined scores. Figure 4.2 shows the overall workflow of model combination for Chinese KB retrieval. Given a Wikipedia document d and a query q, the final interpolation score is

$$
\begin{aligned}
& score(d,q) = \sum_i \alpha_i \cdot score_i(d,q) \\
& s.t. \ \sum_i \alpha_i = 1
\end{aligned}
\tag{4.1}
$$

where $score_i(d,q)$ is the original score assigned by an IR model and α_i is the weight of the model tuned on the development dataset.

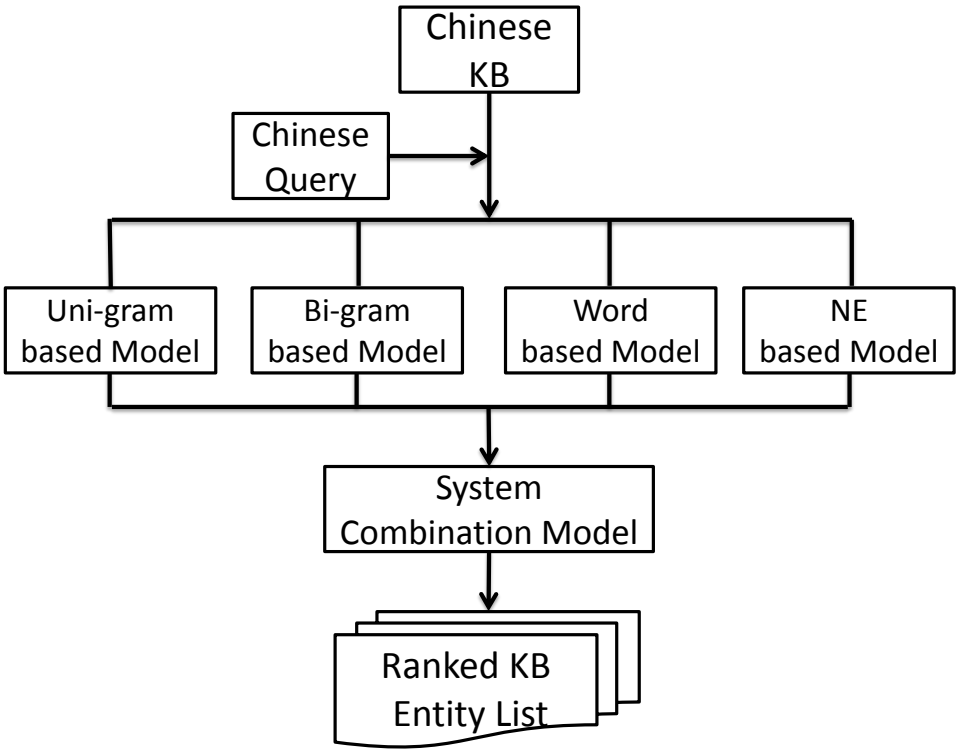

Figure 4.2.. Linear interpolation of retrieval models searching on Chinese Wikipedia.

The top ranked article returned by combined model is considered as a candidate reference to each query. If a mapping from the article title to a KB entry is found, the mapped KB ID is set as the linked ID, otherwise the query is regarded as a NIL entry. In the following section, we will describe the method to cluster NIL queries into different reference clusters.

4.4. Entity Clustering

The major problem of EL is name disambiguation. Linking a query to a KB (or Wikipedia) entity can be treated as name entity disambiguation with Wikipedia as the reference inventory. For those entities without KB linking (a.k.a. NIL entities), NIL clustering is used to automatically group query mentions with the same reference together so that queries within a cluster refer to the same target (sense). The NIL clustering task is similar to the

people name disambiguation task in SemEval-2007 [6], which consists of clustering a set of documents that mention an ambiguous person name according to actual reference targets. The NIL clustering introduces more types of NEs, e.g. location and organization, and bring into addition obstacle of resolving name variations.

The first step of clustering is to group entities with identical names into one coarse group. Based on observations on development data, we assume that all the entities in the same coarse group share identical strings, so we ignore rare cases of different references to entities (e.g. queries "Ford Motor Co." and "Ford" refer to the same company) and cross-lingual reference (e.g. the queries "Hyderabad" and "海得拉巴" in Chinese).

Entities within a coarse group can represent different senses of the entity, such as *Washington* means a person, a city or a state under different contexts. The next step is to scatter them into different fine-grained sense clusters, which are considered as final NIL clusters. We utilize the bag-of-words (BOW) feature from surrounding passages of entity mentions from background document to represent the intended sense of the entity.

An important issue in clustering is to determine the number of NIL sense clusters. Since the number varies from entity to entity it is difficult to train an adaptable clustering model for all entities. A single background document for a query mention does not offer sufficient information to help disambiguate its containing mention. Instead of clustering background documents, we seek more relevant documents retrieved from source collections. Those large amount of documents can be used to determine a significant distribution of word occurrences for each fine-grained sense respectively. As in the KB retrieval model, Indri is called to retrieve top 1000 documents by searching with each query string. Those top relevant documents formalize as a larger document collection for queries with identical surface strings. We use hierarchical clustering algorithm to cluster relevant documents, build the partition of disjoint clusters by cutting the hierarchy, and assign query mentions to most similar sense cluster.

Hierarchical clustering algorithms are either top-down or bottom-up, called agglomerative and divisive clustering respectively. We cluster relevant documents with the Hierarchi-

cal Agglomerative Clustering (HAC) algorithm with a single linkage, which has shown effectiveness on clustering ambiguous person names [125].

HAC [4] initially assigns each document to its own cluster, and a pair of clusters are iteratively merged to form a hierarchy which provides a view of the semantic sense of the entity at different levels of cluster-wise similarity. The merging of pairwise clusters is determined by the combination similarity, which are defined on two criteria: the measure of distance between document vectors including Euclidean distance, squared Euclidean distance, Manhattan distance, maximum distance, cosine similarity etc.; [5] and the linkage criterion which specifies the cluster similarity as a function of inter-similarity between documents from different clusters. Some common strategies lead to single linkage clustering, complete linkage clustering, group-average clustering and centroid clustering. [62]. The single linkage clustering specifies the pairwise similarity of clusters as that of their most similar members. Figure 4.3 visually illustrates that the nearest pair of nodes is taken as the most similar ones, whose similarity determines the similarity of clusters.

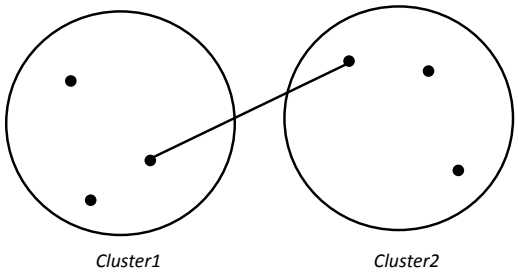

Figure 4.3.. The demonstration of single linkage criterion for cluster similarity used in HAC algorithms.

We transfer the hierarchy into disjoint clusters by cutting it regarding different specification of final clusters [62]. We specify the number of clusters, or number of documents per cluster to determine the cutting point that produces corresponding results. Each document is represented as a BOW vector d of TFIDF values. Based on the validation on development data, we use two different strategies for cutting a hierarchy into a set of flat clusters which will be described in Section 4.6.1.

[4] We use the implementation in scipy http://www.scipy.org/
[5] http://en.wikipedia.org/wiki/Hierarchical_clustering

Each cluster is considered as an accumulation of relevant terms with respect to single entity sense, therefore the background document sharing more terms with a cluster is most likely to share the same sense. The centroid μ of a sense cluster $\mathcal{C}$ can be seen as an approximation of its semantic context,

$$\mu_{\mathcal{C}} = \frac{1}{|\mathcal{C}|} \sum_{d \in \mathcal{C}} d, \tag{4.2}$$

thus a mention occurring in the context similar to the centroid is likely to relate to the same sense as the cluster $\mathcal{C}$.

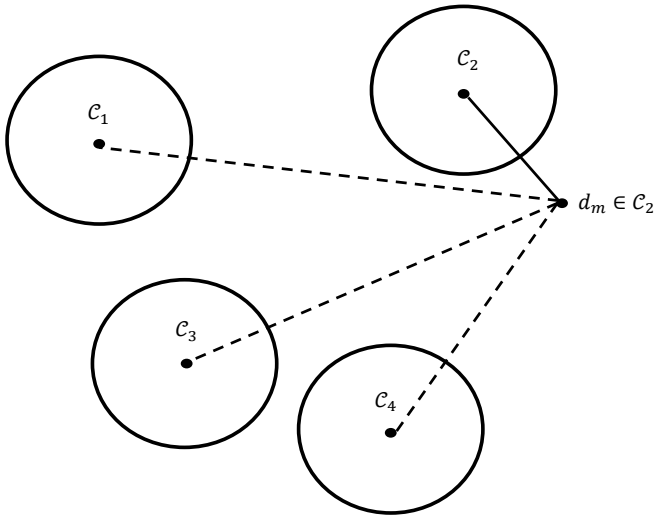

Figure 4.4.. The assignment of clusters based on the distance between a document and each centroid.

We assign the query to the nearest sense cluster by measuring Euclidean distance between its document and each cluster centroid in the vector space as depicted in Figure 4.4. Given a query mention m and background document d_m, its cluster is set as follows,

$$\arg\min_{\mathcal{C}} \|d_m - \mu_{\mathcal{C}}\|_2 = \arg\min_{\mathcal{C}} \sqrt{\sum_i \left(d_m^i - \mu_{\mathcal{C}}^i\right)^2} \tag{4.3}$$

4.5. Comparison of Cross Lingual and Monolingual EL

The performance of Chinese EL system is highly influenced by the mapping from Chinese Wikipedia articles to the English KB. This is mainly due to scale limitation of Chinese Wikipedia and insufficient cross-lingual mappings between English and Chinese Wikipedia. Plenty of disambiguation information is lost during mapping, i.e English Wikipedia provides a disambiguation page mentioning 15 articles named *Denver* while only one Chinese page is about *Denver in Colorado*. [6]

An alternative to Chinese EL approach is to use CLIR approaches, which discover the representations of a meaning in multiple languages by query translations using various MT systems and resources, and then search the English KB directly with English translations.

We translate each query and its expanded terms into English. To minimize the influence of MT ambiguity and errors, we utilized multiple translation strategies:

- **Translation Dictionary** created from interlingual hyperlinks in Chinese and English Wikipedia Pages, such as ”German” [7] corresponds to “德国” [8];
- **Phrase Table** extracted from the LDC parallel Chinese to English NE list [9];
- **N-Best Translation Phrase** of Chinese queries and NEs from documents generated by a Statistical MT (SMT) system [111];
- **Online Translation** of queries from the Google translation [10] .

None of these translations are perfect for all queries, therefore we use the union of all these translations as queries to search English KB. The combination of translated queries can reduce the influence of incomplete translations and provide multiple forms of translated queries, such as the person name “王建民”, Google only manages to translate the last

[6] See zh.wikipedia.org/wiki/Denver and en.wikipedia.org/wiki/Denver_(disambiguation)
[7] http://en.wikipedia.org/wiki/Germany
[8] http://zh.wikipedia.org/wiki/%E5%BE%B7%E5%9B%BD
[9] http://www.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2005T34
[10] translate.google.com

name “Wang”(王), while Wikipedia and LDC dictionaries return two correct translations “Chien-Ming Wang” and “Wang Jianmiing”. The co-occurrence of correct translations also reinforces their importance in the retrieval model, such as the query “最高法院 ” (Supreme Court) , both Wikipedia and Google results are right, while LDC dictionary contains several items on *Supreme Court* in different countries as all NEs containing “最高法院” are selected. The shared translated phrase “Supreme Court” gets more weight in the LM for CLIR. The N-best translation table is the supplement to other translation resources.

For each Chinese query, we create a BOW collection T of English queries using all those translations, e.g.,

T(“安德森”) = {“Anderson”, “Anderson”, ‘Anderzen” , “Andersen”, “the Anderson”, “Mrs Anderson”, . . . }.

For an English token t_e in the translated query T_i of a Chinese query Q_c , $P(t_e|T_i)$ is a relevance language model (LM) [51] which is estimated over all $T_i \in T$.

$$P(t_e|T) = \sum_{T_i \in T} P(t_e|T_i)P(T_i|Q_c), \tag{4.4}$$

where $P(t_e|T_i)$ is calculated by maximum likelihood estimation with Dirichlet smoothing.

$$P_\mu(t|T_i) = \frac{c(t;T_i) + \mu P(t|\mathcal{C})}{\sum_t c(t;T_i) + \mu}, \tag{4.5}$$

where $c(t;T)$ denotes the count of t in T. $P(t|\mathcal{C})$ is the general collection probability calculated from English Wikipedia to model the probability of unseen words.

The translation probability from Chinese to English $P(T_i|Q_c)$ is defined as the phrase-to-phrase translation probability if T_i is generated from the SMT system, otherwise it is set to 1 as T_i is the result of dictionary matching or Google translation, which are assumed to produce perfect translations of the query.

We improve LM-based retrieval by including inference networks [95]. Our inference network

includes the target entity and chunks extracted from passages with a mention of the entity in the reference document. We use $P(t_e|T)$ as weights in the inference network of retrieval model, which makes the retrieval process more robust against noisy expansion terms.

Micro-averaged accuracy introduced in Section 2.1.2is used to compare the effectiveness of different retrieval strategies on the end-to-end performance of the system. The best CLEL performance is achieved by combining Google and SMT translation results, listed in Table 4.2. The results in Table 4.3 show the evaluation of monolingual retrieval on Chinese KB. The overall accuracy of monolingual retrieval model is 5% better than that of CLIR. The overall linked and NIL accuracy of monolingual model are both better than those of cross-lingual model.

	All Entities	**PER**	**ORG**	**GPE**
Overall	0.65	0.67	0.65	0.62
in-KB	0.46	0.25	0.45	0.58
NIL	0.86	0.88	0.8	1

Table 4.2.. Performance (Micro-Average Accuracy) of **cross-lingual** EL strategy for Chinese queries on TAC 2011 **development** data.

	All Entities	**PER**	**ORG**	**GPE**
Overall	0.7	0.71	0.86	0.52
in-KB	0.51	0.45	0.67	0.47
NIL	0.91	0.84	1	1

Table 4.3.. Performance of **monolingual** EL strategy for Chinese queries on TAC 2011 **development** data.

The monolingual strategy generates balanced performances for different types of queries, however the cross-lingual model achieves much worse results on person queries. This is due to some Chinese entities especially person names that do not receive any proper translation. Although we incorporate multiple translation resources, some queries do not get a proper translation at all such as “四通集团” in Table 4.4.

The monolingual model is not without its shortcomings. For the GPE queries, the cross-lingual model achieves 10% better than monolingual model as dictionary-based methods

Chn. Query	Wiki Dict.	Google	LDC NE	N-best
王建民	'Chien-Ming Wang'	'Wang'	'Wang Jianming'	N/T
最高法院	'Supreme Court'	'Supreme court'	'supreme court of justice' 'supreme court of andorra' 'court of justice'	'the supreme court' 'the supreme court of' 'the highest court'
安德森	'Anderson'	'Anderson'	'Anderzen', 'Andersen' 'Anderssen', 'Andersson' 'Andson', 'Anderson'	'the Anderson' 'Mrs Anderson'
嘉禾集团	N/T	'golden harvest group'	N/T	the group of 嘉禾
四通集团 (*SiTong Group*)	N/T	'Stone Group'	N/T	'four work groups' 'four poetry group'
乔斯	'Jos'	'Jose'	'Jose', 'Jos', 'Trzos', 'Csausz', 'Chaus', 'Cios'	'Jose', 'Josh' 'Georgey', 'Georges' 'sorry.george'

Table 4.4.. Sample translations generated with different methods. **N/T** stands for "No Translation".

and Google translation generally performs better on translating location names, which are less ambiguous and highly covered by translation vocabularies. Taking the difference of performances into consideration, we adopt the monolingual retrieval strategy for Chinese entities.

4.6. Results

4.6.1. Cross-lingual Entity Linking

Our three runs submitted to the CLEL task adopt the same monolingual document retrieval method described in Section 4.3. Those runs are involved with different clustering strategies. The baseline run **lsv1** only employs the coarse clustering, whereas the runs **lsv2** and **lsv3** employ HAC clustering on the result of the baseline and cut the clustering hierarchy into flat clusters in different ways. The configurations are as follows:

- **lsv1:** simply clustering monolingual entities with the same literal names together. Cross-lingual clustering is not considered.
- **lsv2:** when making clusters of top relevant documents, at most 2 sense clusters are set for each identical entity.
- **lsv3:** 50 documents in each sense cluster for each identical entity.

Table 4.6 demonstrates that HAC clustering can improve the clustering performance over the baseline by 8.7% and 4.6% in configuration **lsv2** and **lsv3**. Considering the performance of different clustering configurations in Table 4.5, the HAC methods used in **lsv2** and **lsv3** contribute to nearly no improvement over the baseline **lsv1**.

Table 4.7 summarizes the performances of Chinese and English EL results. Unlike most teams of former evaluations we do not employ the KB node candidate generation method. We retrieve all articles from the Wikipedia collection, which is larger than the KB collection. The low linked KB accuracy of both languages indicates that it is hard for retrieval-based

Run	Prec.	Recall	F-score
lsv1	0.514	0.581	0.545
lsv2	0.515	0.577	0.544
lsv3	0.519	0.579	0.547

Table 4.5.. Performance of different system configurations on the 2011 cross-lingual entity linking **evaluation** data.

Run	Prec.	Recall	F-score
lsv1	0.514	0.567	0.539
lsv2	0.547	0.632	0.586
lsv3	0.512	0.628	0.564

Table 4.6.. Performance of different system configurations on the 2011 cross-lingual entity linking **development** data.

		All	PER	ORG	GPE
English Entities	Overall	0.51	0.46	0.58	0.46
	in-KB	0.34	0.41	0.39	0.21
	NIL	0.80	0.84	0.81	0.77
Chinese Entities	Overall	0.65	0.57	0.77	0.62
	in-KB	0.44	0.30	0.59	0.47
	NIL	0.80	0.70	0.86	0.98

Table 4.7.. Micro-averaged accuracy of Chinese and English entities on TAC 2011 **evaluation** data.

methods to find the right reference from a large collection of documents even with query expansion. Regarding each entity type, the performances on English GPE entities and Chinese PER entities suggest the limited domain adaptation of the retrieval methods. It is better to adopt specific modules for different entity types. The title-to-ID mappings produce fair NIL accuracy for both languages as we expected, however it is too strict to verify only top one entry and simply ignore desirable targets in other top rankings.

4.6.2. Result Analysis

By the comparison of retrieval performances on training and evaluation queries in Table 4.8 and 4.9, it can be seen that all different segmentation-based retrieval models perform worse on the evaluation dataset than on the training dataset, and both the overall linked and NIL accuracies decrease on the evaluation dataset.

The performance of retrieval model on person and organization queries in the evaluation dataset is better than those in the development dataset, while performance on GPE queries in training dataset is inferior to that in the development dataset, which is also revealed by comparing Chinese retrieval results in Table 4.3 and 4.7. This is due to person and organization queries in the evaluation dataset is less than those in development dataset. GPE queries in the development dataset contains more NIL queries, so even the NIL accuracy of GPE in evaluation dataset is less than that in development dataset, the overall accuracy of GPE is still more than that on development dataset. Appropriate weighting of terms is critical for linear interpolation. Those parameters tuned on the development dataset do not apply suitably on the training dataset. The generalization ability of our monolingual models need to be further improved.

		All	**PER**	**ORG**	**GPE**
Unigram	Overall	0.622	0.604	0.820	0.447
	in-KB	0.465	0.435	0.684	0.377
	NIL	0.797	0.691	0.918	1.000
Bigram	Overall	0.662	0.672	0.862	0.447
	in-KB	0.468	0.459	0.671	0.377
	NIL	0.878	0.782	1.000	1.000
POS	Overall	0.646	0.628	0.783	0.532
	in-KB	0.498	0.459	0.595	0.473
	NIL	0.811	0.715	0.918	1.000
NER	Overall	0.587	0.556	0.683	0.532
	in-KB	0.508	0.494	0.595	0.473
	NIL	0.676	0.588	0.745	1.000

Table 4.8.. Micro-averaged accuracy of Chinese entities on TAC 2011 **development** data with retrieval models on different indexing units.

		All	PER	ORG	GPE
Unigram	Overall	0.606	0.526	0.732	0.596
	in-KB	0.395	0.213	0.560	0.452
	NIL	0.771	0.690	0.821	0.933
Bigram	Overall	0.641	0.568	0.787	0.596
	in-KB	0.423	0.249	0.613	0.459
	NIL	0.811	0.736	0.876	0.917
POS	Overall	0.612	0.535	0.771	0.559
	in-KB	0.429	0.285	0.653	0.423
	NIL	0.755	0.667	0.832	0.875
NER	Overall	0.501	0.410	0.569	0.571
	in-KB	0.485	0.335	0.700	0.487
	NIL	0.514	0.450	0.502	0.767

Table 4.9.. Micro-averaged accuracy of Chinese entities on TAC 2011 **evaluation** data with models on different indexing units.

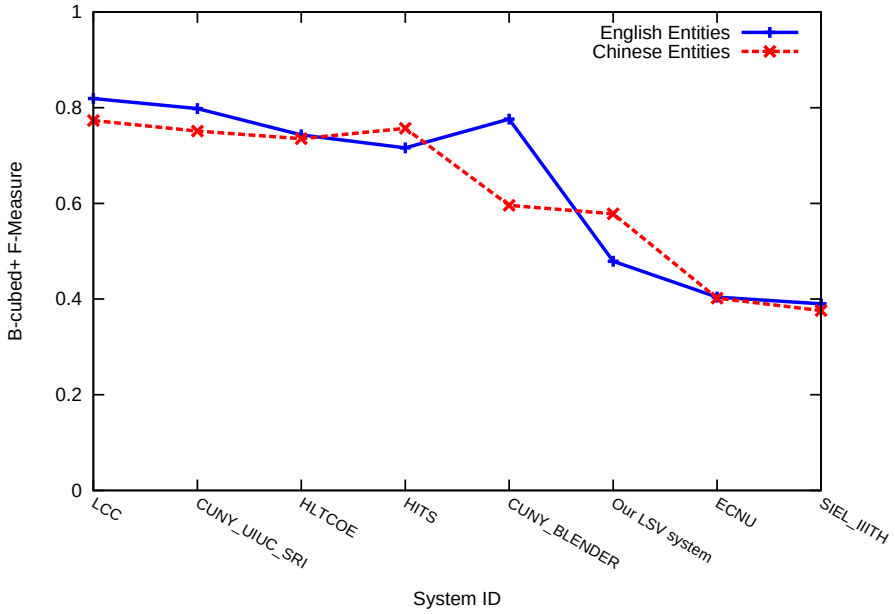

Figure 4.5.. Performance of TAC KBP 2011 CLEL Systems on English and Chinese queries.

Our system ranked 6th out of 12 teams in the final evaluation of CLEL task in 2011. The overall performance of CLEL systems are summarized in Table 4.10. Figure 4.5 demonstrates the performance of Chinese and English queries separately. Our cross-

System ID	B–Cubed+ F-measure
LCC	0.788
CUNY_UIUC_SRI	0.766
HLTCOE	0.738
HITS	0.727
CUNY_BLENDER	0.654
Our LSV system	**0.547**
ECNU	0.403
SIEL_IIITH	0.386

Table 4.10.. Overall performance of cross-lingual entity linking systems in TAC KBP 2012.

lingual performance on Chinese queries is almost comparable to that of CUNY_BLENDER's system, which has integrated SMT and cross-lingual entity similarity methods. As we focused on dealing with the Chinese part of CLEL evaluation, the comparable medium performance of English EL leaves lots of room for improvement in the future work.

4.7. A Simple Chinese Entity Linking Model

In this section, we introduce a new generative EL model for Chinese queries. The new system architecture is displayed in the Figure 4.6. Comparing with the original architecture displayed in Figure 4.1, the new system deprives the document retrieval component, and includes new candidate generation and candidate ranking components. The motivation of new candidate generation method is to achieve higher precision of NIL queries, and provide higher recall of linked entities for generative candidate ranking model to achieve higher linked accuracy.

4.7.1. Candidate Generation

Our CLEL system makes use of IR to select entity candidates from Chinese KB. Regarding the major candidate generation strategies used in TAC's top EL systems, we propose the new candidate generation method for Chinese queries. For a query mention, we attempt

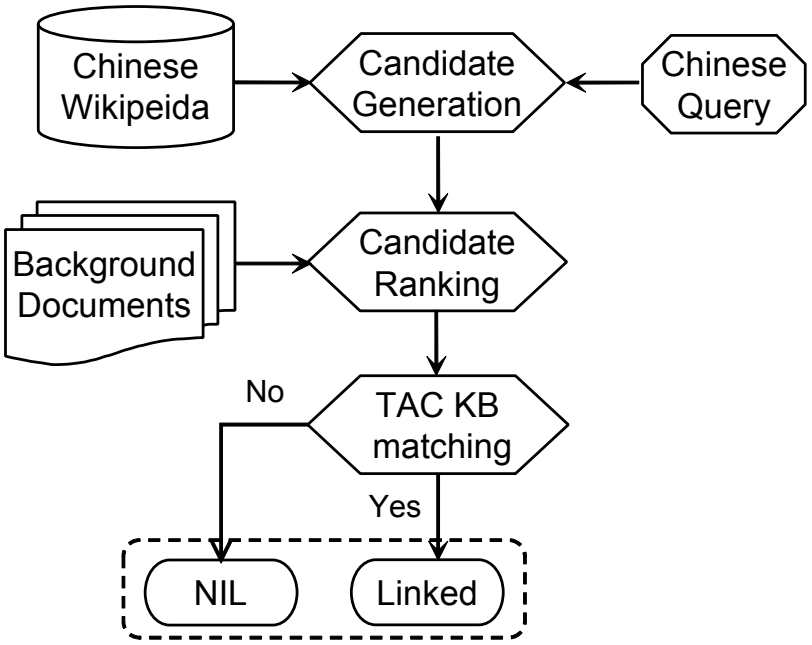

Figure 4.6.. New CLEL system architecture with new candidate generation and candidate ranking components.

to identify every possible correct Chinese KB entity in the following steps.

1. **Matching query strings with potential referents**. The expanded query set is checked against article names, redirect page names and hypertext anchors in disambiguation pages of all Wikipedia titles, therefore Wikipedia entries containing expanded queries are taken as general referents.

2. **Candidate Postfiltering.** If a mention string appears at the begin and end at a referent name, the referent is included in the candidate set; otherwise the referent is removed from the candidate set.

3. **NIL checking.** All referring Wikipedia titles are validated whether the corresponding entries exist in English KB. After validation, if the candidate set contains no KB entity, the query is directly tagged as NIL. These non-empty candidate sets are to be sorted by the candidate ranking model.

The first step makes sure that every query gets all the possible candidates from Chinese Wikipedia, so the candidate set is ensured with high recall for correctly linked entities. As Chinese text has no spaces as word boundaries and Chinese word boundary is highly ambiguous, the first step generates many candidates and makes it hard to pinpoint with the ranking method. such as the query “根特” (Ghent) matches the entity “艾根特·沙乌” (Agent Sawu); the query “摩西” (Moses) matches “狄摩西尼” (Demosthenes). These entities are irrelevant with respect to the query although they contain the query. The candidate filtering can prune these irrelevant entities.

We test the candidate generation method on the development dataset. The recall for NIL entries is much higher than that of the retrieval model in the old system, meanwhile a considerable part of irrelevant candidates is removed. 87 out of 616 linked queries receive no candidates mainly due to the following reasons:

- Absence of KB queries, such as the query “达尔文” (Darwin) refers to “the city of Darwin”, which has no Chinese counterpart.
- Transliteration variations, such as the query “瑟琳娜” (Serena) refers to “Serena Williams” with Chinese counterpart titled “塞雷娜·威廉姆斯”. The transliteration of “Serena” is 塞雷娜(Pinyin: *Sai Lei Na*) instead of 瑟琳娜 (Pinyin: *Se Lin Na*).
- Acronym for named entities, i.e., Chinese literature usually refers to foreign location names after its first occurrence with first character of their Chinese words, such as the query “巴民族权力机构” (Palestinian National Authority), where 巴 stands for “巴勒斯坦” (Palestinian).

4.7.2. Generative Cross-Lingual Entity Linking Model

The EL task heavily relies on knowledge. In fact, the skeletal procedure of a general EL system can be summarized as follows: given a query mention and a set of words from background documents, a technique is applied which makes use of one or more sources of knowledge to associate the most appropriate entities with mentions and contextual

words. Knowledge sources can vary considerably from collections of raw documents to more structured resources in Wikipedia.

Han and Sun [33] proposed a generative entity mention model, which pinpoint the linked English entities by learning heterogeneous knowledge, including popularity knowledge, name knowledge and context knowledge. For the CLEL task, we utilize a simplified entity mention model. Our model captures the entity popularity from both English and Chinese Wikipedia, and adopts a simplified and effective method for modelling knowledge from background documents. The new model achieved a significant increase in linking performance over our baseline system for Chinese queries.

A basic entity ranking problem is set up as follows: given a name mention m to be linked with KB and a set of entities $e_1, e_2, \ldots, e_n$, which are output from a generative EL model. The top-ranked entity is the exact entity to link. Typically the mention and entities are expressed in the same language, such as English. Although the query of CLEL is in Chinese, we convert it into Chinese monolingual EL by deriving the Chinese KB from Chinese Wikipedia.

The uncertainty whether an entity is linked to a query mention is modelled by the uncertainty associated with inferring the evidence from Wikipedia and background documents. Under this assumption, $P(m|e)$ can be estimated indirectly using a generative model in the following way:

$$\begin{aligned} e = \arg\max_e \frac{P(m,e)}{P(m)} &\propto \arg\max_e P(m,e) \\ &\approx \arg\max_e P(e)P(c|e) \end{aligned} \tag{4.6}$$

P(e) is estimated with global statistics from Wikipedia, and $P(c|e)$ is learned from the local contexts in background documents.

Popularity-based Features

Section 3.1 describes the linking structure in Wikipedia. Every Wikipage is annotated with anchor links when it is cited the first time in other Wikipages. Intuitively the more often a entry is linked, the more popular it is. This popularity knowledge is very helpful for linking the conventional and unambiguous entities, such as famous cities and company names.

The popularity distribution $P(e)$ is inferred with the maximum-likelihood estimation of occurrence of e as a link in all Wikipedia pages. We count the overall number M of anchor links and sum up the occurrence of e is an anchor link as $Count(e)$, then

$$P(e) = \frac{Count(e) + 1}{M + N} \tag{4.7}$$

where N is the count of normal pages in Wikipedia. The probability of unseen mention is captured by the add-one discounting method.

Chinese Entity	English Entity	Popularity(Chn.)	Popularity(Eng.)
伊丽莎白二世	Elizabeth II	2.7366×10^{-05}	7.222×10^{-05}
伊丽莎白一世 (英格兰)	Elizabeth I of England	8.2099×10^{-06}	3.356×10^{-05}
伊丽莎白·泰勒	Elizabeth Taylor	3.619×10^{-06}	1.116×10^{-05}
茜茜公主 [a]	Empress Elisabeth of Austria	1.942×10^{-06}	1.503×10^{-07}

[a] Redirect from the page titled 伊丽莎白·阿马利亚·欧根妮 (Elisabeth Amalie Eugenie)

Table 4.11.. Sample candidate entities for the query “Elizabeth”, and their popularity in Chinese and English Wikipedia.

The popularity distribution of Chinese referents can be interpreted with statistics from both English and Chinese Wikipedia. To obtain $Count(e)$ of a Chinese entity e from English Wikipedia, we fetch the English counterpart $\mathcal{E}(e)$ of e using interlingual title mapping, and then calculate $Count_{en}(\mathcal{E}(e))$ in English Wikipedia, using it as $Count(e)$.Table 4.11 shows the popularity of entities derived from Chinese and English Wikipedia.

4.7.2.1. Context-based Features

Clearly $P(e|m)$ is context dependent. The distinct entity that m is assumed to link to, depends on the context in which it occurs. Given the following sentences,

1. 英国 女王 伊莉莎白 二世 在 白金汉宫, 以 茶点 款待 二千 名 英国 儿童 。

(*British Queen Elizabeth II treated 2,000 British children with refreshments at Buckingham Palace.*)

2. 19 世纪 奥地利 皇后 伊莉莎白 的 珍珠 钻石 胸针 "西西 之 星"

(*"The Star of Sisi" pearl diamond brooch of Elizabeth, Empress of Austria in 19th Century*)

In the popularity distribution shown in Table 4.11, the entity "Empress Elisabeth of Austria" has a smaller probability than "Elizabeth II". This may be a reasonable model for describing contemporary query mentions, but it may be inaccurate for specific entities about Austro-Hungarian Empire in 19th century, to which the entity "Empress Elisabeth of Austria" is highly related. The context model $P(c|e)$ is used to model the context knowledge.

Regarding the content in the first sentence, the probability $P(c|$**Elizabeth II**$)$ shall be higher than $P(c|$**Empress Elisabeth of Austria**$)$, while vice versa for the second sentence. $P(c|e)$ is cast as the relevance between an excerpt containing mentions from background documents and Wikipedia pages on the entity e. We use the following steps:

1. Extract background excerpts consisting of summarized texts and local texts, equivalently the headline and first sentence of the background document, and 50 words in the contextual window surrounding query mentions, respectively.
2. Both the Wikipedia documents and background excerpts are viewed as a BOW of terms and NEs, and we disregard all words and phrases except nouns and NEs when formatting the vector space model. The cosine similarity between background document and Wikipedia pages is utilized as the estimation of $P(c|e)$.

4.7.3. Evaluation

From Table 4.12 and Table 4.13, we can see only popularity models (from either English or Chinese Wikipedia) can achieve better results. It is similar to the observation in TAC EL 2010 [42]: "a naïve candidate ranking approach based on web popularity alone can achieve 71% micro-averaged accuracy." The context model along achieves less linked accuracy but better NIL accuracy. This is due to mismatching of non-target Wikipedia pages, which contain many occurrences of query tokens and contexts. The context model leads to worse clustering performance, which proves that it recognizes many linked entities as NIL in ranking results. The joint model of popularity and context knowledge can achieve a balance of linked and NIL accuracy, with circa 5% increasing of micro averaged accuracy, as well as at least 4.5% clustering B-Cubed+ F-score over individual models. The new model achieves a significant performance improvement over our old system, and can reach the second position in Chinese EL comparing with participant systems in TAC 2011.

		All	**PER**	**ORG**	**GPE**
English Popularity	Overall	0.710	0.548	0.864	0.802
	in-KB	0.617	0.339	0.707	0.789
	NIL	0.783	0.657	0.945	0.833
Chinese Popularity	Overall	0.716	0.560	0.859	0.807
	in-KB	0.615	0.330	0.747	0.771
	NIL	0.794	0.681	0.918	0.892
Chinese Context	Overall	0.710	0.665	0.850	0.629
	in-KB	0.538	0.511	0.647	0.502
	NIL	0.845	0.745	0.955	0.925
English Popularity + Chinese Context	Overall	0.742	0.618	0.875	0.794
	in-KB	0.657	0.507	0.713	0.746
	NIL	0.809	0.676	0.959	0.908
Chinese Popularity + Chinese Context	Overall	**0.752**	0.637	0.871	0.805
	in-KB	**0.680**	0.529	0.747	0.763
	NIL	**0.807**	0.693	0.935	0.900

Table 4.12.. Micro-averaged accuracy of the Generative EL Model strategy for Chinese queries on TAC 2011 **evaluation** data.

	B–Cubed+ Precision	B–Cubed+ Recall	B–Cubed+ F-score
English Popularity	0.605	0.706	0.652
Chinese Popularity	0.618	0.711	0.661
Chn. Context	0.608	0.674	0.639
Eng. Popularity + Chn. Context	0.643	0.725	0.682
Chn. Popularity + Chn. Context	0.656	0.731	**0.691**

Table 4.13.. Clustering evaluation of generative model EL strategy for Chinese queries on TAC 2011 **evaluation** data.

4.8. Conclusion

We have developed a CLEL system for TAC KBP 2011, which uses a parallel monolingual architecture mainly consisting of a document retrieval and an entity clustering module. We show the feasibility of using Chinese Wikipedia as the KB to connect cross-lingual knowledge and avoid the propagation of translation errors to the retrieval module. To reach a better solution for the CLEL problem, we propose a simple generative EL model, which utilizes new entity generation and reranking model to improve the EL performance.

Part II.

Learning with Crowdsourced Annotations

Introduction

Linguistic annotation is fundamental and indispensable to various tasks in natural language processing (NLP), information extraction (IE) and information retrieval (IR), since several supervised and semi-supervised machine learning methods trained on annotated data have been successfully introduced and applied in those fields and achieved state-of-the-art performances. Many open-domain evaluations such TREC and TAC evaluations, and professional linguistic resource annotation societies, such as Linguistic Data Consortium (LDC) [11] have devoted a lot of effort to collecting, annotating, and sharing linguistics resources in order to meet the gradually increasing needs for annotated data for various tasks.

Although those resources have promoted the research and applications in IR and NLP, those traditional in-house annotation procedures suffer from some inherent shortcomings. Training annotators takes a lot of effort and is expensive. The process of annotation usually takes a long time to accomplish, which makes it hard to support emergent tasks. On the other hand the access of annotation is not very convenient. Users have to participate in evaluations or become paying members to be granted access to specific datasets. The expenditure is not suitable in some situations. For individual research, the cost of training, supervising and managing a network of skilled and responsible annotators often outweighs the value of completing the annotation. Therefore we need alternate data collection and annotation paradigms, which are more flexible, faster and cheaper.

In recent years the emergence of *crowdsourcing* provided considerable opportunities for

[11] http://www.ldc.upenn.edu/

fast distributed resource collection and creation. According to Wikipedia, crowdsourcing is a neologism defined as

> *the act of taking a task traditionally performed by an employee or contractor, and outsourcing it to an undefined, generally large group of people or community in the form of an open call.*

The usage of crowdsourcing can occur both online and offline. The traditional annotation can be viewed as offline crowdsourcing with comparably small groups of annotation, while the online crowdsourcing addressed in this thesis takes advantage of aggregating crowd intelligence. The development of online crowdsourcing platform, such *Amazon Mechanical Turk* (AMT) [12] and *Crowdflower* [13] make it possible for requesters to publish and distribute their tasks to the large crowd of online workers and get work done effectively and efficiently.

In this part of the dissertation, first we will focus on using AMT to create a corpus of supporting passages for list question answering. Due to the variety of worker's expertise and task complexity, the results of crowdsourcing annotation are not perfect. To improve the quality of annotation, we also employ various methods to learn gold standard-annotation from noisy AMT annotations. We train learning to ranking models with the enhanced annotation and achieve start-of-the-art performance.

Note that most of the work in Chapters 6 and 7 has been published as [116, Xu and Klakow, 2010]. Some of the work in the Chapter 8 has been published as [117, Xu and Klakow, 2012].

[12] http://www.mturk.com
[13] http://crowdflower.com/

5. Background on Question Answering

Asking questions has been one of the main activities for people to learn and acquire knowledge. With the development of internet and search engine techniques in recent years, people can easily access massive information. Coming with the epochal phenomenon of "information explosion", it becomes more difficult for many people to easily manage and effectively organize information, which leads to the problem of *information overload*.

The development of automatic Question Answering (QA) systems offers a possible solution to *information overload*. The goal of QA is to extract answers for natural language questions. It can help us to filter and summarize knowledge from tremendous electronic information. To answer a question such as "List capital of European countries", a QA system can properly comprehend the meaning of the question and automatically give the list of answers. Advanced QA technologies, which deeply learn the breadth of relevant content to more precisely extract and justify answers, "can help support professionals in critical and timely decision making in areas like compliance, health care, business intelligence, knowledge discovery, enterprise knowledge management, security, and customer support". [28]

As a research field, QA is primarily concerned with developing theories, principles, algorithms and systems to help a user find correct answer(s) to a question from a collection of text documents. Recently lots of research activities have emerged to solve the QA problem from the perspective of Information Retrieval (IR) and Natural Language Processing (NLP). The surge was initiated by QA tracks of TREC (Text REtrieval Conference)[1] for English

[1] http://trec.nist.gov/.

language in 1999 [106]. In the coming eight years increasingly powerful systems have been developed and tested on QA benchmarks. Following this trend, other evaluations have been organized to promote research, innovation, and development of multilingual and cross-lingual QA. There are CLEF (Cross Language Evaluation Forum)[2] for European languages [77] and NTCIR (NII-NACSIS Test Collection for IR Systems) for east Asian languages[3] [30]. Those conferences defined some specific types of questions with restricted forms of answers which are easy to evaluate.

Over the decades, those evaluations have contributed to solve critical techniques in QA, and generated a resurgence of research topics for IR and NLP communities. Researchers have developed a suite of automatic QA technologies, to deal with various questions on specific knowledge in restricted domains and open questions about universal topics which are accessible to human beings.

This chapter focuses on the creation of question and answer datasets for the development and research in QA. We do not intend to give a though survey on research topics in QA, however we will give a general overview of underlying techniques in QA and how our following work on passage ranking can fit in the overall framework of QA system.

Section 5.1 briefly reviews the early history of QA. Section 5.2 introduces the chronicle of QA evaluation tracks. Section 5.3 presents the architecture of our latest QA systems for TAC 2008 to demonstrate how the overall system and underlying models work.

5.1. Early History of Question Answering

Over the decades, many QA approaches and systems have been proposed, studied and tested. Research in QA can be dated back to the 1960s. The earlier systems mainly deal with restricted domain by searching structured knowledge in a database.

The first successful system — BASEBALL [32]was constructed to answer specific questions

[2] http://www.clef-initiative.eu/
[3] http://research.nii.ac.jp/ntcir/

about base games in the American league. The database-based QA system actually worked quite promising by modern QA evaluation standards as they employed natural language interface to database system [5], which enabled users to type requests expressed in some natural languages, while back-end system included linguistic analysis to convert the natural language question into a canonical from, which the database management system can process and find the matched answer.

Another success restricted-domain system was LUNAR [110], which was developed "*to enable a lunar geologist to conveniently access, compare, and evaluate the chemical analysis data on lunar rock and soil composition that was accumulating as a result of the Apollo moon missions*". During a demonstration at the second annual lunar science conference in 1971, LUNAR successfully answered 78% of the 111 questions on the topic of moon rock. Many QA systems have been developed for answering questions in specific domains. More examples and comprehensive reviews can be found in [5, 40].

Although those approaches have higher accuracy in specific domains, they can only process natural language from limited fields. It takes lots of effort to construct ontology and update database to extend those systems to new domains without authoritative and comprehensive resources. Consequently researchers began to exploit more powerful systems for automatically extracting answers from unstructured texts. It leaded to more challenging open-domain QA.

Open domain QA, which aims at coping with questions on nearly every topic, relies on general ontologies and captures knowledge from unstructured documents. One of the early open-domain system — MURAX [49] answered general-knowledge questions based on an online encyclopedia. They investigated the feasibility of combining NLP and IR methods for QA with unrestricted texts. They also presented the pipeline architecture. Both methodologies have been widely utilized as standard approaches for building QA systems. The typical pipeline architecture relies on four main steps:

1. **Question Analysis**, to extract key phrases from questions and the semantic type of expected answers;

2. **Document Retrieval**, to retrieve relevant documents from a large collection of corpora;
3. **Passage Retrieval**, to choose the best answer passages from documents;
4. **Answer extraction**, to extract the final answer(s) from answer-bearing passages.

The development of NLP technologies has lead to significant improvements of QA methods and systems. In the last 15 years, the NLP community has witnessed a significant shift from the use of manually crafted systems to the employment of automated statistical methods. Consequently lots of sophisticated NLP approaches were introduced to boost the performance of QA approaches. One of the most important events was the large-scale QA evaluations hold by TREC during 1999 and 2008, which have greatly accelerated the research and development in open-domain QA. In the following sections we will introduce successive QA evaluations in TREC and our participating QA systems.

5.2. Question Answering at TREC

Comparing and evaluation QA systems is extremely difficult due to the variety of question sets, document collections and evaluation methods adopted. The international TREC workshops and evaluation tracks were held every year since 1990. They organize annual evaluations and provide the infrastructure for large-scale comparative evaluation of several IR tasks. In 2008, NIST separated QA tracks from TREC and merged with other evaluations to initiate the new TAC workshops, which focus on large-scale NLP evaluations. The following QA resources were released every year to the participants for their system development.

- various classes of test questions;
- several corpora of document collections from which the answers shall be extracted;
- ranked documents retrieved by running IR engines on the indexing of corpora;

- judgement files of submissions indicating the correctness of answers;
- correct answer patterns and document IDs.

Table 5.1 chronicles the series of QA evaluations sponsored by TREC and TAC. The QA datasets and evaluations have been governed by various guidelines, which have changed over successive workshops. We now briefly review and discuss the evaluation tracks held between 1999 and 2008. A review of the first five TREC QA tracks can also be found in Voorhees and Harman [100].

TREC 1999 and 2000: Original Tasks The task definition of TREC 1999 [106] is that given a document collection and a test set of questions, participants are required to return a document ID and text snippets (in 50 bytes or 250 bytes), containing an answer to the question. Those questions asked for short fact-based answers, such as "*How many calories are there in the a Big Mac?*" and "*Where is the Taj Mahal?*" Each participant submitted the response of his system, afterwards human assessors checked each response and decided whether the answer snippet did contain an answer to the questions regarding the contexts in the document.

Given a set of judgements, the original evaluation metric was Mean Reciprocal Rank (MRR). It is the average of the reciprocal of the rank at which the first correct answer appears. Let Q be the question collection and r_i be the rank of first correct answers to question i or 0 if no correct answer is returned.

$$MRR = \frac{\sum_{i=1}^{|Q|} \frac{1}{r_i}}{|Q|} \tag{5.1}$$

The task definition in TREC 2000 [107] was identical to that in 1999. It introduced more documents in the collection and selected 'real' questions gathered from two query logs [4].

[4] http://encarta.msn.com and http://www.excite.com/

	99	00	01	02	03	04	05	06	07	08
Answer Type	Answer Snippet			Exact Answer						
Document Collection	TREC	TREC + TIPSTER		AQUAINT					Blog06 + AQUAINT	Blog06
Question Type	Factoid		Factoid +List	Factoid+List +Definition						Opinion
Question in Series	NO			YES						
Num. of Question	200	693	500	500	500	351	530	567	515	177
Evaluation Measure	MRR			CWS	F-measure					

Table 5.1.. Task definition of TREC/TAC QA tracks.

TREC 2001: List Questions The main task [101] was similar as before. The modification was that questions might have no answers from the document collection. The *list* task was introduced as the second task, which was much harder than main task due to the uncertain size of answer set. A single document could contain multiple answer instances, and the identical instances might be present in multiple documents. Systems did not get credits for submitting duplicate answers. List QA results were evaluated using accuracy, the proportion of number of distinct responses to the total number of answer instances.

TREC 2002: Exact Answers TREC QA 2002 [102] repeated the main and list tasks, and made the crucial requirement that participating systems shall extract answers instead of answer snippets. An answer-string must contain a complete, exact answer item and nothing else. The support, correctness, and exactness of answers was in the opinion of the assessors. They assigned one of four possible judgements to an item:

- **incorrect**: the answer-string does not contain a correct answer item;
- **unsupported**: the answer-string contains a correct answer item but the document returned does not support that answer item;
- **non-exact**: the answer-string contains a correct answer item and the document supports that answer item, but the string contains more than just the answer item (or is missing bits of the answer item);
- **correct**: the answer-string is exactly a correct answer item and that answer item is supported by the document returned.

Each submission only got credits for correct answers.

Another difference is that system should predict confidence scores for answers and rank the list of answers, hence systems were given more credits if they ranked correct answers higher than incorrect ones. In this case, MRR was disadvantageous as the evaluation metric for QA systems as it gave not credit to systems which retrieved multiple non-duplicate correct answers. TREC 2002 proposed the new evaluation metric — Confidence

Weighted Score (CWS).

$$CWS = \frac{\sum_{i=1}^{|Q|} \frac{\text{number correct in first } i \text{ ranked answers}}{i}}{|Q|} \tag{5.2}$$

TREC 2003: A Combined Task The main task [103] was presented as a combined task, including three classes of questions:

- **Factoid questions** ask about a aspect about a item and requires a short named entities as answer. For example, "*What company acquired IMG in 2004?*" The only answer is "Forstmann Little & Co.".
- **List questions** request for a set of instances as answers. For example, "*Who are members of the board of the IMG?*" has several answers.
- **Definition questions** give additional detailed information which might be of interests to users. [5] It is usually answered by longer texts and evaluated against defined answer nuggets. For example, "Introduce the company IMG."

The evaluation of list questions was based on the collection of answers returned for each questions. Let S be the number of correct instances given by human assessors, D the number of correct *distinct* answers returned by a system, and N the total number of answers returned by the system. Therefore the precision of system was given as $P = \frac{D}{N}$ and recall as $R = \frac{D}{S}$. The evaluation score — F-measure was defined as

$$F = \frac{2 \times P \times R}{P + R} \tag{5.3}$$

The definition task was a new task. It asked interesting facts about a topic, such as "*Who is Colin Powell?* or "*What is mold?*". The answer snippets for definition questions were judged against a set of information nuggets created by human assessors. Each nugget referred to a single atomic piece of information about the given question. For example,

[5] in the QA track, the user is assumed as an "average" adult reader of American newspapers.

seven nuggets were expected to return for the question " *Who was Alexander Hamilton?*":

- *Secretary of the US Treasury*
- *killed in duel with Arron Burr*
- *charged with financial corruption*
- *congress refused to impeach*
- *confessed to adultery*
- *leader of the federalist party*
- *named chief of staff by Washington*

The assessors judged whether a response contained the vital nuggets. The final score of a definition question was evaluated with F-measure.

Let r be the number of vital nuggets returned in a response;
a be the number of acceptable (non-vital but on the list) nuggets returned in a response;
R be the total number of vital nuggets in the assessor's list;
len be of the number of non-white space characters in an answer string summed over all answer strings in the response;

then,

$$\begin{aligned} recall &= \frac{r}{R} \\ allowance &= 100 \times (r + a) \\ precision &= \frac{allowance}{len} \\ F(\beta = 5) &= \frac{\beta^2 \times precision \times recall}{(\beta^2 + 1)precision + recall} \end{aligned} \tag{5.4}$$

The final score for evaluation a system was reported as an a combination over all questions from those three tasks, which was defined as

$$FinalScore = \frac{1}{2} \times FactoidScore + \frac{1}{4} \times ListScore + \frac{1}{4} \times DefScore. \quad (5.5)$$

The final score emphasized on the traditional factoid questions, which dominated the largest proportion of questions. The weights for the list and factoid subtasks were made large enough to attract participation in those subtasks.

TREC 2004–2007: Series of Questions, Blog Data TREC QA 2004 [104] started to arrange testing questions into groups to formate several question series. Each series is defined with a topic (target), such as the topic *220* named "*International Management Group (IMG)*". Each question in a series was about a facet of the topic, which covered on people, organization and other entities. TREC 2005 [105] questions extended topics on events. TREC 2006 [24] made minor change of the task guideline that each system should return the most up-to-date answers. The time stamp of answer-bearing document was used as the time of answer.

The main task of TREC 2007 [23] kept the tradition of QA. To evaluate how systems processed diverse genres of unstructured texts, new blog data were added to the document collection. It also adjusted the judgement of answer correctness for factoid questions by including the following criteria:

- **locally correct**: the answer string consists of exactly a correct answer that is supported by the document returned, but the document collection contains a contradictory answer that the assessor believes is better;
- **globally correct**: the answer string consists of exactly the correct answer, that answer is supported by the document returned, and the document collection does not contain a contradictory answer that the assessor believes is better.

The evaluation score was a combination of individual series' scores, each of which was an

average of scores of those questions in the series:

$$FinalScore = \frac{1}{3} \times FactoidScore + \frac{1}{3} \times ListScore + \frac{1}{3} \times OtherScore \quad (5.6)$$

TAC 2008: Opinion Questions In 2008, NIST made a major change of tracks — splitting QA track from TREC and coupling with Document Understanding Conference (DUC) [6] in the new Text Analysis Conference (TAC) [25]. The objective of TAC QA was identical to that of TREC QA: participants were required to retrieve the answers to a set of questions.

The major change went to question types. TAC questions series asked for people's opinions about a particular target, which were retrieved from blog data. There were two types of questions — rigid list questions and squishy list questions, which asked for exact instances of a specified type, and answer snippets within certain length respectively. The question series *1047*: "*Trader Joe's*" comprised the following questions:

- **RIGID** *Who likes Trader Joe's?*
- **SQUISHY** *Why do people like Trader Joe's?*
- **RIGID** *Who doesn't like Trader Joe's?*
- **SQUISHY** *Why don't people like Trader Joe's?*

The evaluation of rigid list questions was the same as list questions, while the evaluation of squishy list questions was the same as definition questions. The final evaluation score was based on scores of individual series, which were an average of scores of two tasks:

$$FinalScore = \frac{1}{2} \times RigidListScore + \frac{1}{2} \times SquishyListScore \quad (5.7)$$

5.3. Brief Overview of *Alyssa* QA System

Our group has developed a statistically inspired open-domain QA system (*Alyssa*) to participate in TREC/TAC QA evaluations. We give a brief overview how the Alyssa system

[6] http://duc.nist.gov/

works in this section.

For TAC 2008 QA track, our system Alyssa was modified according to new requirements of opinion questions. Figure 5.1 shows the architecture of Alyssa. Alyssa 2008 defined two streams — an adapted version of our factoid stream in 2007 [85] and a completely new stream which was designed for the questions asking for bloggers. Blogger question detection classified questions into two types using a rule-based approach. The questions asking for bloggers run through both the main stream and the blogger stream whereas the other questions only run through the main stream.

The main stream comprised eight main modules: Question Analysis, Semantic/Polarity Question Typing, Query Construction and Expansion, Document Retrieval, Sentence Retrieval, Sentence Annotation, Answer Extraction, and Answer Validation. We first performed a linguistic analysis to generate structured presentations of questions. The results of syntactic parsing and NE tagging were used later for answer extraction. The semantic type of a question is determined in a separate step called semantic question typing. We adopted a model using support vector machines (SVM), which produced a higher classification accuracy both on the sample questions provided by NIST for TAC 2008 competition and our own set of opinion questions. Beside the semantic question typing, the polarity of opinion questions was determined by the polarity question typing component. A query was formulated from the question with results from those analysis.

Following query construction, we applied query expansion techniques based on *Google* and *Wikipedia*. The expanded query was run against document retrieval on the Blog06 corpus. The dynamic document fetching [85] determined the number of retrieved documents according to the question type. The sentence retrieval component retrieved relevant sentences based on language modeling.

The opinion sentence retrieval module selected opinionated sentences from the retrieved sentences. Sentence polarity classification was applied to retrieve opinionated sentences in order to classify the sentences as positive or negative. The sentences with the same polarity typing as their question were chosen for further processing.

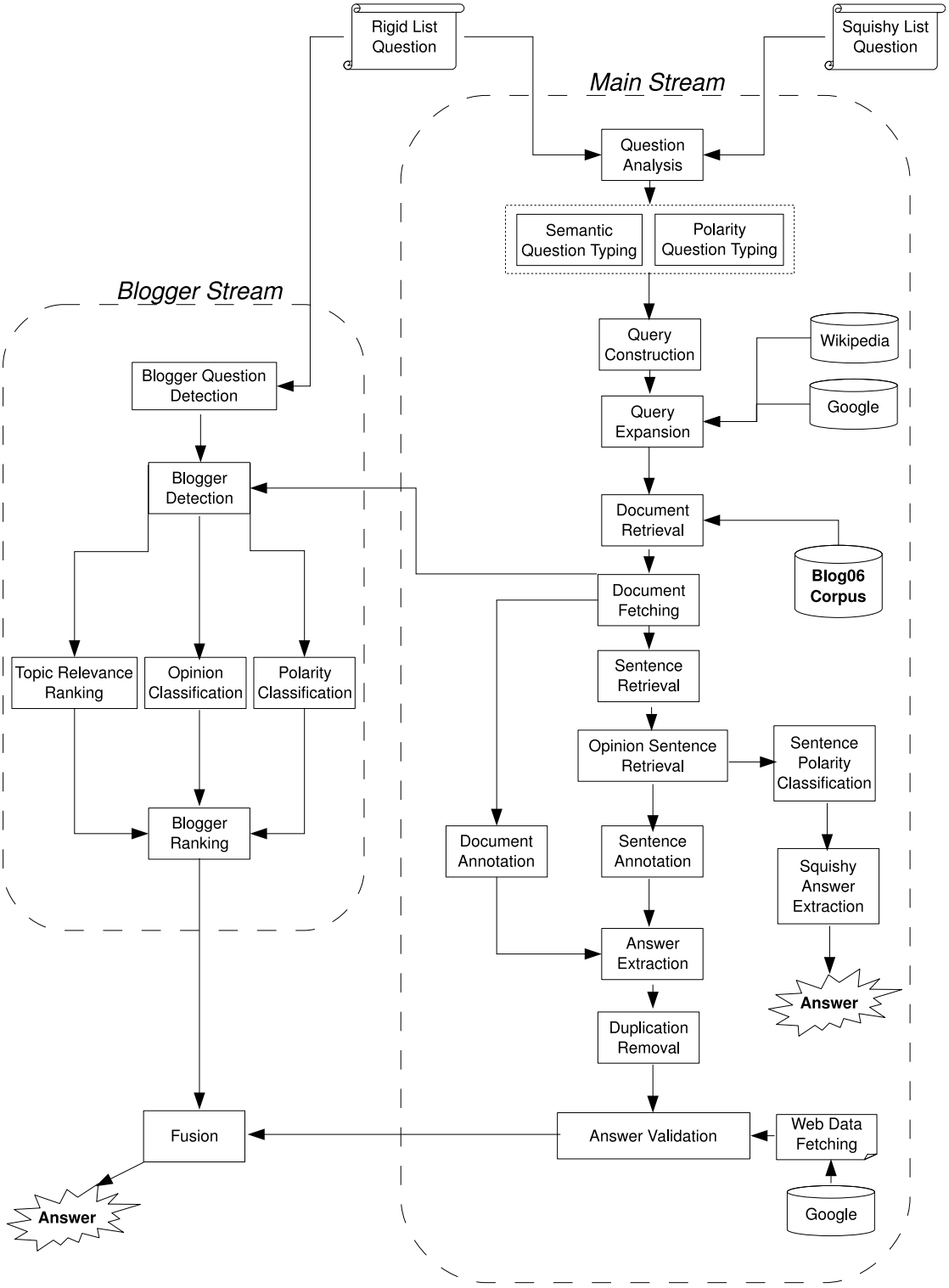

Figure 5.1.. The archtecture of *Alyssa* Question Answering System.

Squishy list questions did not require exact short answers, we thus directly employed squishy answer extraction to generate answers. Answering rigid list questions requires two types of linguistic processing. If the question asked for an NE from the entertainment domain, we automatically annotated retrieved documents with NEs of the corresponding types. Otherwise, only the opinionated sentences with the correct polarity are annotated.

After the extraction of candidate answers from the annotated documents or sentences, duplication removal was applied. Our web-based answer validation component re-ranked the resulting list of unique candidate answers as the final answers to rigid list questions.

The blogger stream of Alyssa followed after document fetching of the main stream. In the blogger stream the retrieved documents underwent blogger detection to split the document into smaller segments and find the author/blogger of each segment. Each segment was assigned three scores estimated by three different components: topic relevance ranking searching for the relevant segments to the question, opinion classification computing the degree of opinionatedness, and polarity classification measuring how much the polarity of a segment overlaps with the polarity of its question. The interpolation of scores from individual components was assigned to each segment. Finally the segments were ranked in the blogger ranking according to those scores. In the fusion module, the result of blogger questions was merged with the output of the main stream which created a unique list for blogger rigid list questions.

For more details about the components and evaluations of our systems, refer to our participation papers on QA tracks [22, 85, 109].

6. Crowdsourcing for Paragraph Acquisition and Selection for QA

6.1. Introduction

As described in Chapter 5, question answering (QA) is a challenging task for the information retrieval (IR), information extraction (IE) and natural language processing (NLP) communities. The TREC QA evaluation tracks [1] cover a broad range of techniques, which involve with learning from annotation data. The demand for annotation data is urgent and varies by sub-tasks in QA.

This chapter studies how to create corpus for the passage ranking task in answering list questions, which have multiple answers. As a practical representative, we choose the task of creating supporting passages for list questions. We run annotation tasks via Amazon's Mechanical Turk and recruit online workers to distinguish the passages that answer a question from unsupportive passages. We extract an intermediate corpus comprising pairs of questions and answer-bearing passages from relevant documents, and then collect multiple online judgements on whether each passage supports its given question. To improve the annotation quality, we introduce methods to learn the true labels from multiple annotations by learning annotator credibility and task difficulty. After the annotation collection and enhancement, the new listQA corpus consisting of supporting passages is beneficial for various QA tasks, such as passage retrieval and answer extraction. The

[1] http://trec.nist.gov/data/qa.html

methodology of crowdsourcing annotation can also generalize to other task-oriented annotation for various NLP tasks.

6.2. Amazon Mechanical Turk

Mechanical Turk [2] is a crowdsourcing internet marketplace hosted by Amazon. Barr and Cabrera [8] proposed that crowdsourcing provides so-called *artificial artificial intelligence*, which enables human intelligence to be easily and programmatically accessed and incorporated into software applications. With Aamazon's Mechanical Turk (AMT) as the intermediary platform, developers or researchers can submit their micro-tasks, get them done swiftly, approve or refuse completed tasks and combine the results into their own applications or research. Meanwhile the internet-scale active human workforce completes chosen tasks and receives payments for their approved work.

For the requesters, AMT provides various interfaces to design, publish and manage their tasks. Each Task usually consists of a bunch of micro Human Intelligence Tasks (HITs) (e.g., each pair of question and passage in our task), so that AMT can distribute individual tasks as micro-tasks to be simultaneously processed. The HITs running on AMT mostly belong to tasks which are easily solvable for human intelligence, nevertheless challenging for artificial intelligence, such as labelling objects in a picture, translating texts, transcribing podcasts, writing summarization for articles. The requesters are provided with web user interfaces to monitor, reject and accept HITs while workers get paid for their completed and accepted HITs. Figure 6.1 shows the snapshot of a sample HIT preview to a worker via AMT. In this HIT, workers are asked to translate Chinese words in English.

Figure 6.2 illustrates the worker interaction Model of AMT — Search-Continue-RapidAccept-AcceptPreview (SCRAP) model proposed by Heymann and Garcia-Molina [36]. MTurkers can first `Search` or `Browse` some summary snippets of HITs. Workers can then get more information on interesting tasks by taking a `Preview` of concrete assignments in

[2] https://www.mturk.com/

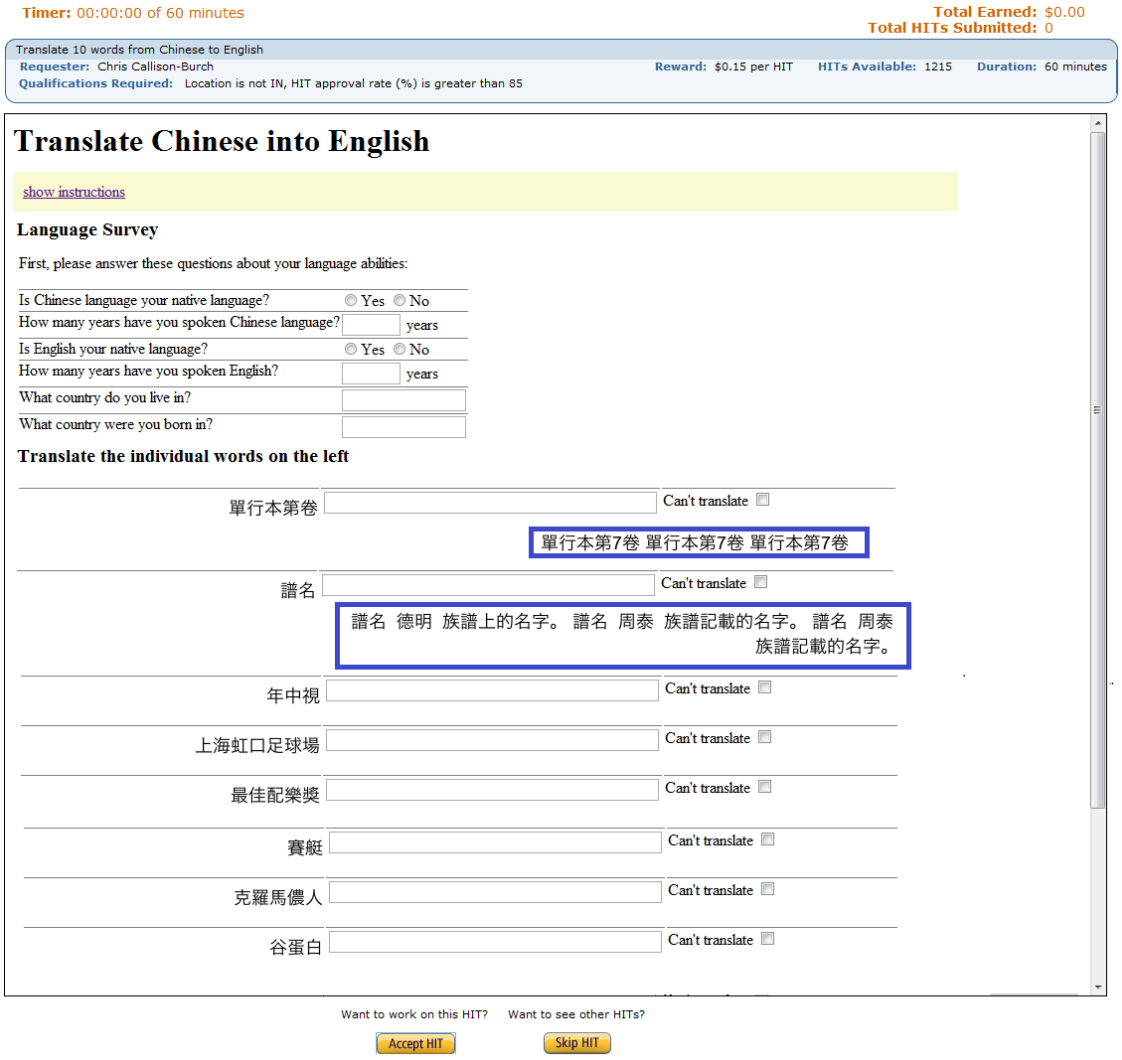

Figure 6.1.. preview of a HIT on phrase translation at AMT. On the top left shows the time that a worker has spent on the HITs, and the right top presents the number and value of submitted HITs so far. Below is the detailed information of HITs including the requester, reward per HIT, number of HITs, working duration and required qualifications (MTurkers are from non-US location and achieve minimum HIT approval rate of 85%). The following pane demonstrates the main interaction interface of the HIT. The language survey provides the assessment of workers' language proficiency; MTurkers need to translate Chinese phrases into English. When they click on the input field, a excerpt (highlighted in frame) is shown as explanatory reference.

HITs. The workers can choose to skip the current HIT to view others, so he can review and understand the overall tasks, then decide whether to take action of `Accept` the HIT and work on assignments. During the period of actual processing on assignments, there are two actions — `Continue` and `RapidAccept`. A worker can continue completing assignments that was accepted but not submitted or returned. `RapidAccept` allows a worker to keep working assignments in a HIT group without pause of previewing it first. Lastly, in each step the worker can always return to a previous state.

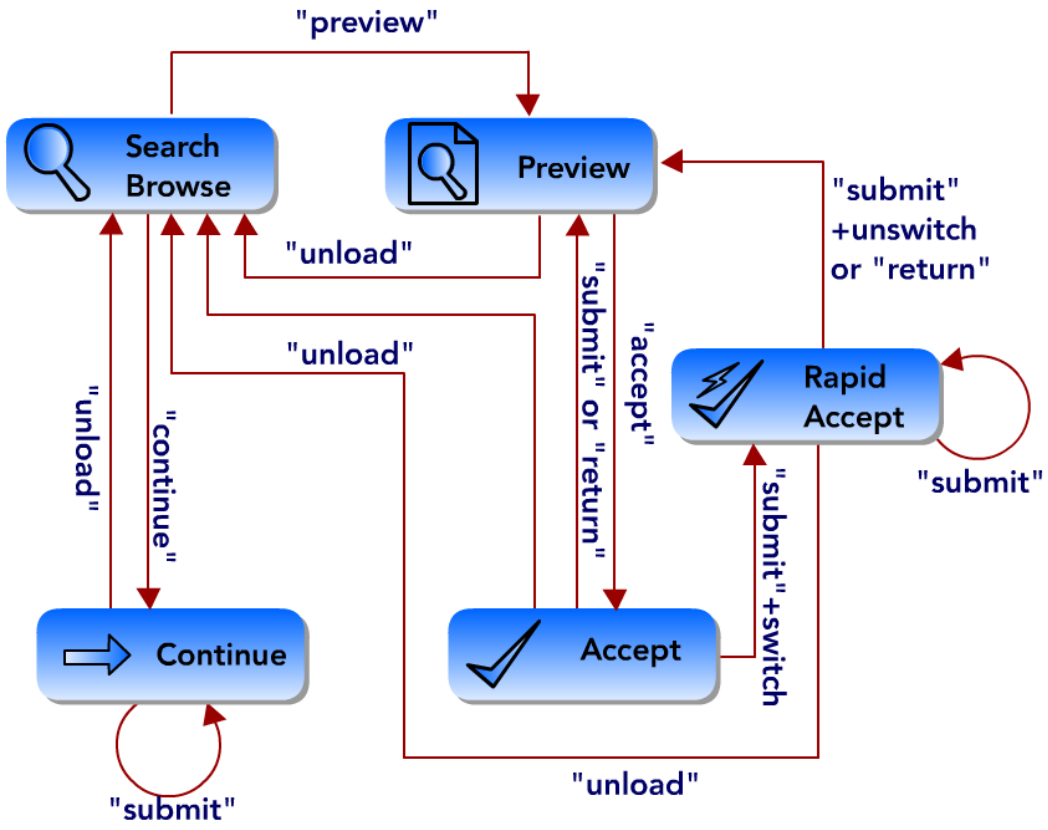

Figure 6.2.. Search-Continue-RapidAccept-Accept-Preview (SCRAP) interaction model from Heymann and Garcia-Molina [36].

With those promising interaction with a large amount of workers, AMT creates tremendous opportunities for using real-time human computation for a range of diverse tasks. The five key concepts of AMT are *HITs*, *workers*, *qualification*, *assignments* and *requesters*.

The requesters are individuals or organizations who want their work done through human computation. Those people who work on HITs to earn money are called workers (a.k.a. MTurkers).

For the requesters, the workflow on AMT begins with designing and publishing HITs. Each HIT is originally rendered from a pre-defined HTML template by loading input data to be annotated by workers. Real-time HIT pages display text and multimedia data to MTurkers, and utilize standard HTML form elements to collect their responses as annotation or evaluation to the task. If requesters want to restrict their HITs to a specific group of workers, they can require that MTurkers meet certain criteria or pass certain qualifications before pursuing on HITs. Requesters can reject the low-quality responses or incompetent workers, so they do not pay for them, or even block substandard MTurkers from participating in the future HITS.

The HIT being processed by a MTurker is called an assignment to him. Requesters can change the amount of workers for an individual HIT. Each worker can work on multiple assignments but never work on duplicate assignments. This *plurality* feature offers an important and powerful tool to assess the quality of responses. With the responses of multiply workers, requesters can select the major agreed response as the final result.

AMT provides powerful and versatile methodology to support harvesting crowd wisdom. Requester can create their tasks by simply using the representative templates via web user interface, describing their structure by using XML, or utilizing markup or scripting languages such as HTML and JavaScript to add multimedia and interactive elements.

When the HIT is running or complete, developers can view and manage their tasks via web service, requester APIs or command line tools. The Mechanical Turk Sandbox [3] is served as a practical simulated environment so that developers can freely publish and test HITs from the perspective of both requesters and workers. Requesters can view the summary of task submission, assignment and completion trough web interface. When a HIT group is completed, requesters can view or download online stored results. Requesters and

[3] https://requester.mturk.com/developer/sandbox

workers can also communicate with each other on HITs.

In recent years, there has been an increasing interest in collecting human intelligence via AMT for NLP and IR tasks. Snow et al. [88] made the first attempt of collecting labels for several NLP tasks. Their experimental analysis demonstrated that accumulation of judgements from multiple non-expert MTurkers can reach the quality of annotations from expert annotators. They managed to get 140+ hours worth of human effort from MTurkers to produce 21,000 labels for just over $25. Other successful applications vary from those simple *decision* tasks such as relevance evaluation for the information retrieval and extraction systems [2, 70, 10], to advanced data *creation* and *enhancement* tasks, such as creating high quality translations by aggregating and editing multiple translations [120], and producing highly parallel data from video actions for paraphrase evaluation [13].

The advantages of AMT for data evaluation and collection include:

A scalable, low-cost Workforce. It connects to more than 500,000 workers from 190 countries [4], who have diverse and independent skill sets and capabilities. Most HITs on AMT costs a few cents, therefore requesters can submit numerous assignments with less cost.

Fast turnaround. With the availability of massive online workforce, multiple workers process HITs simultaneously, therefore even those numerous HITS can be completed very fast.

Flexibility. The easy micro-payment and versatile design system offers convenient functionalities to scale up tasks and attempt ad-hoc experiments from a variety of perspectives with easy control of budgets.

All annotation methodologies are not perfect and noise-proof. Despite the scalability, efficacy and flexibility of AMT annotation, there is increasing concern that AMT suffers from some deficiency due to the nature of crowdsourcing. Because of the anonymous population of MTukers, it is very hard to prevent incompetent or irresponsible works,

[4] https://requester.mturk.com/tour

and more seriously extensive artificial results are produced by internet bots. In most cases MTurkers only complete few assignments of a HIT group, so the consistency of annotated results is not guaranteed. For those reasons, the online annotation consists of considerable erratic results. Recent developments in crowdsourcing research have heightened the following questions:

- How to assure the quality of MTurker-generated decision and content?
- How to learn better assessments by learning from those information?

We will explain out attempt to solve the first question in Section 6.3 and provide detailed analysis of a possible solution to the second in Chapter 7.

6.3. Experiment Design

6.3.1. TREC data sets

Chapter 5 reviewed the activities and developments of research work in QA. For a given question, a participating system is required to provide one answer for the *factoid question* or a list of distinct answers for the *list question*. An eligible response for submission includes an answer string and the identification of a answer-bearing document. When all submissions are complete, TREC recruited human assessors judge the correctness of the answer to the question based on the content from the document.

Each year after the completion of system evaluation, TREC would release a collection of gold standard answers for the question set, including answer patterns, which are regular expressions for matching answering strings derived from correct answers in all responses from participating systems. Answer patterns are used to match answer-bearing passages from relevant documents. Although this method matches passages which only contain the answer but no enough contextual evidence to answer the question, the aggregation of crowdsourcing judgement is used to easily distinguish those false positive passages.

Each answer pattern consists of *question id*, *answer regexps* and *document ids*, the format is listed in Table 6.1. The question 3.3: *In what country was the Hale Bopp comet visible on its last return?* have correct answers including *Australia*, *China*, *Panama* and *United States*. Each of them maps to an answer pattern in Table 6.1. The answer *United States* is correctly found in several documents with identifiers of *XIE19960217.0069, XIE19960105.0039, etc.*. Table 6.2 shows examples of passages from relevant documents matched by answer patterns. The first passage supports its answer "*China*" to question 3.3, while the second does not.

Ques. id	Regexp	Document ID List
3.3	`Australia`	XIE19960321.0254
3.3	`(Chinese|China)`	XIE19960405.0124 XIE19970319.0243 ...
3.3	`Panama`	XIE19970318.0242
3.3	`(United States| America|US)`	XIE19960311.0115 XIE19960409.0120 XIE19960217.0069 XIE19960105.0039 ...

Table 6.1.. Example of Answer Patterns.

Judgement	Passage for annotation
`Supportive`	Hale–Bopp, a newly-discovered extraordinarily large comet in the solar system, has been recently observed for the first time in *China*.
`Unsupportive`	At 11.39 p.m. local time, "only a very thin rim will be seen over the orange face" of the moon, a color that will become brighter in the shadow of the earth, the *Panama* Canal Commission said.

Table 6.2.. Support judgement of passages matched with answer patterns.

Tellex et al. [93] quantitatively evaluated the effectiveness of several passage retrieval algorithms for QA by including the answer patterns in strict and lenient judgement of the passage relevance. The strict scoring determines whether a passage matches the answer pattern and appears in a supporting document, while lenient score requires only pattern matching. These scoring generalized the evaluation metrics for document retrieval to passage retrieval, however more fine-grained supporting passages are needed for further analysis and evaluation. To achieve this goal, we focused on creating supporting passage corpus for list QA, which is much harder to solve than factoid QA and has not been thoroughly studied.

Currently there is no such dataset of question-supporting texts for TREC list question task. The purpose of our work is to contribute to the development of QA systems by providing a new corpus, which include pairs of a question and a passage which supports its containing answer to the question. The applications of IR, IE and NLP techniques in QA will benefit from the fine-grained annotated dataset.

For the factoid question answering, Kaisser et al. [47] constructed the corpus of supporting *sentences* for factoid questions by running annotation tasks via AMT and postprocessed MTurkers' results with the validation of specialists. In contrast to expensive and time-consuming relevance judgement by few assessors [100], AMT offers a web-based solution to quickly and cheaply annotate supporting compact excerpt in the relevant documents. Our work is not only to construct the corpus of supporting passages for list question task, but also to investigate and compare various automatic learning methods to select true annotations. Those learning methods can largely improve the quality of annotation from AMT and save the cost and time of post-processing as in Kaisser et al [47], which makes it possible to implement crowdsourced annotation for large-scale and successive linguistic data annotation.

Following the workflow introduced in Section 6.2, we conduct the data collection in following steps:

1. **Data Generation.** We first use passage bound tags to break each document into several passages. We select passages matched by answer patterns to be candidates, which will be presented together with the question to MTurkers.

2. **HIT design.** The principle of HIT designing is to appropriately organize and present the annotation data so that annotators can easily understand the requirement and execute the annotation. We therefore keep the task description succinct and set up a straightforward user interface. We adjust the qualification and number of MTukers to reach the best preliminary results through several dry runs.

3. **Automatic Selection of True Annotations.** To filter the noisy annotation and im-

prove over the majority voting of multiple annotations, various methods are explored to enhance the quality of annotations.

Details will be introduced in following sections and chapter.

6.3.2. Data and Experiment Setup

To set up a HIT group, we need to prepare the following elements:

1. **Input data**. Input data consists of tuples of *question*,*answer* and *passage*. We generate 2856 question-answer-passage tuples for TREC 2004.
2. **Interface template**. The interface template is an HTML page with three variables indicating a *question*,*answer* and *passage*. All HITs are created from rendering the interface template.
3. **Running HITs.** The input data is uploaded to MTurk. The submission procedure automatically replaces those variables in template with values from input data, and then publish a batch of active HITs as depicted in Figure 6.3.
4. **HIT cost.** The aim of our HITs is to judge whether each passage supports its containing answer(s) to the given question. It is a fairly easy task like most HITs running on AMT. To control the budget, every HIT costs $0.02 and contains one question and two passages, so MTurkers become more familiar with the task by repetitive working on one question. [5]

After several dry runs with various options of assessments, we finally adopt a binary relevance criterion: supporting and non-supporting passages. [6] We also ask the MTurkers to describe task difficulty and provide comments, which may provide specificities of questions in individual assignments.

[5] Most HITs on AMT cost $0.01 each.

[6] We tried different scales for relevance assessment, with partially supporting document, marginally supporting document and uncertain of relevance. The experimental results shows that more options seemed to confuse the workers and resulted in inconsistent judgement.

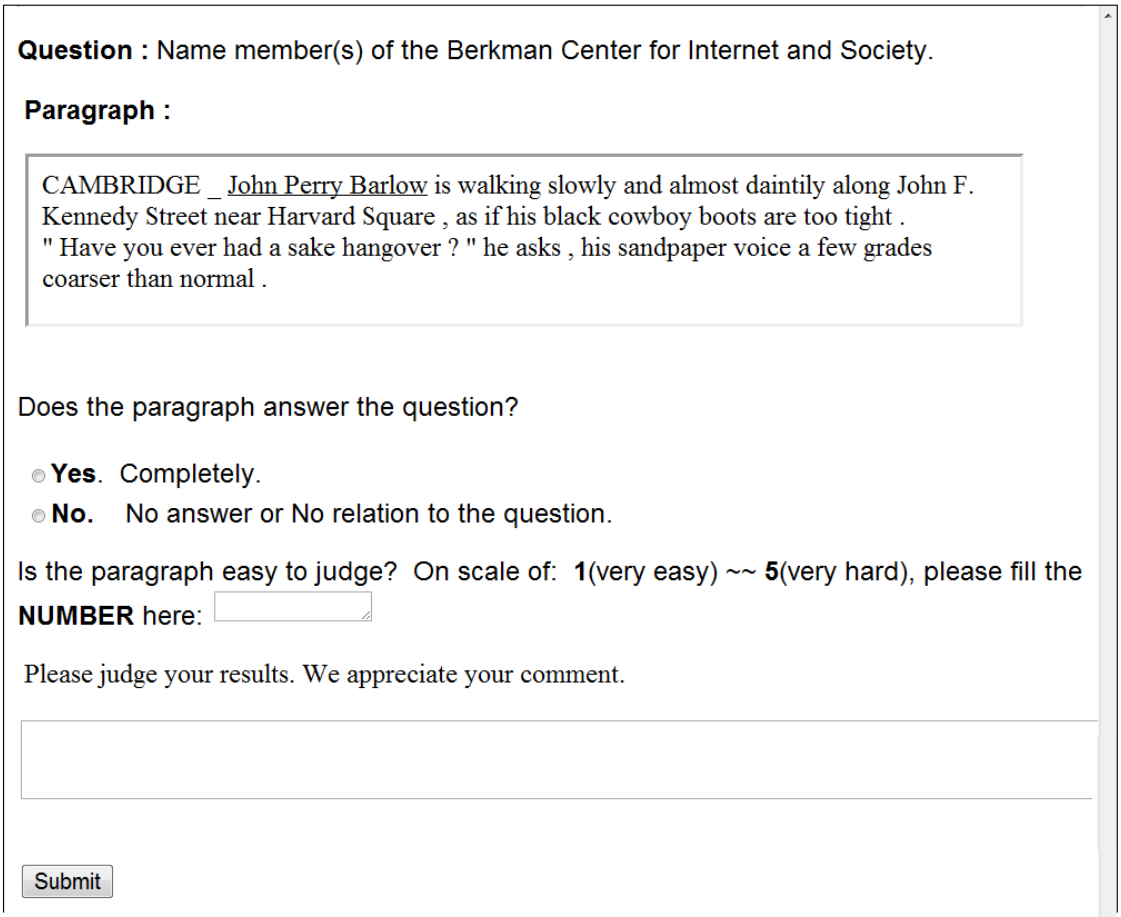

Figure 6.3.. Sample real-time HIT with one pair of question and passage. The matched answers are highlighted with underline. The task instruction is, "*Given a question and a passage, please judge whether the passage answers the question. If the passage answers the question, and the passage contains one answer or more, the correct response is 'Yes'; If not, the response is 'NO'. Please only refer to the passage, don't use common sense.*".

6.3.3. Data Quality Control

In this section we will describe and discuss our efforts on using built-in options for qualification control provided by AMT. We mainly consider two important options: the HIT approval rate and plurality of workers per HIT.

The HIT approval rate provided by AMT is defined as the proportion of individual worker's approved assignments to all his submitted assignments. Regarding the user privacy issue,

Amazon provides no personal information about workers except the identification assigned automatically to MTurkers. The statistics based on MTurker's historical working records can probably indicate the quality of his future work, therefore the HIT approval rate is very important for filtering workers. We set the requirement that MTurkers are qualified to take our assignments by having achieved the minimum HIT approval rate.

The HIT approval rate helps prevent incompetent workers, nevertheless we need additional methods to evaluate the performance of workers on the current task. The measurement for evaluating the quality of MTurkers' work is *annotation accuracy* — the proportion of assignments that are correctly judged by workers according to gold-standard annotations. For the initial run, we set the rate more than 95 (The MTurker has had 95% of his assignments accepted by requesters.) and recruited 3 workers per HIT, and there are 8568 assignments in all. The annotation result (depicted in Dataset A) is very noisy. To minimize the negative effects from the diverse worker expertise and noisy annotation, we therefore increase the approval rate to more than 98 and recruited 5 workers per HIT (totally 14280 assignments), and the final AMT annotation result is denoted as Dataset B.

Based on crowd-annotated results, we manually created the gold-standard annotations to evaluate the quality of work. Among the gold-standard 2856 passages, 1300 passages completely support their containing answer(s) to the given question, while rest 1556 passages are irrelevant or partially relevant to the questions.

Table 6.3 shows the comparison of annotation accuracy from dataset B and dataset A. The single most striking observation to emerge from the data comparison is that accuracy increases by 24.40%, from 49.37% to 73.77%. Table 6.4 demonstrates the inter-annotator agreements, i.e. how often a certain number (Two to Five) of workers agree on the same judgement about one HIT. Figure 6.4 shows the relation between individual worker's accuracy with number of their completed passages.

It is apparent from the comparison of dataset A and B that even though we increase the approval rate, there still exist some spam workers as the points scattering in the right part of both figures, which indicates annotation accuracy of around 50% for binary judgement,

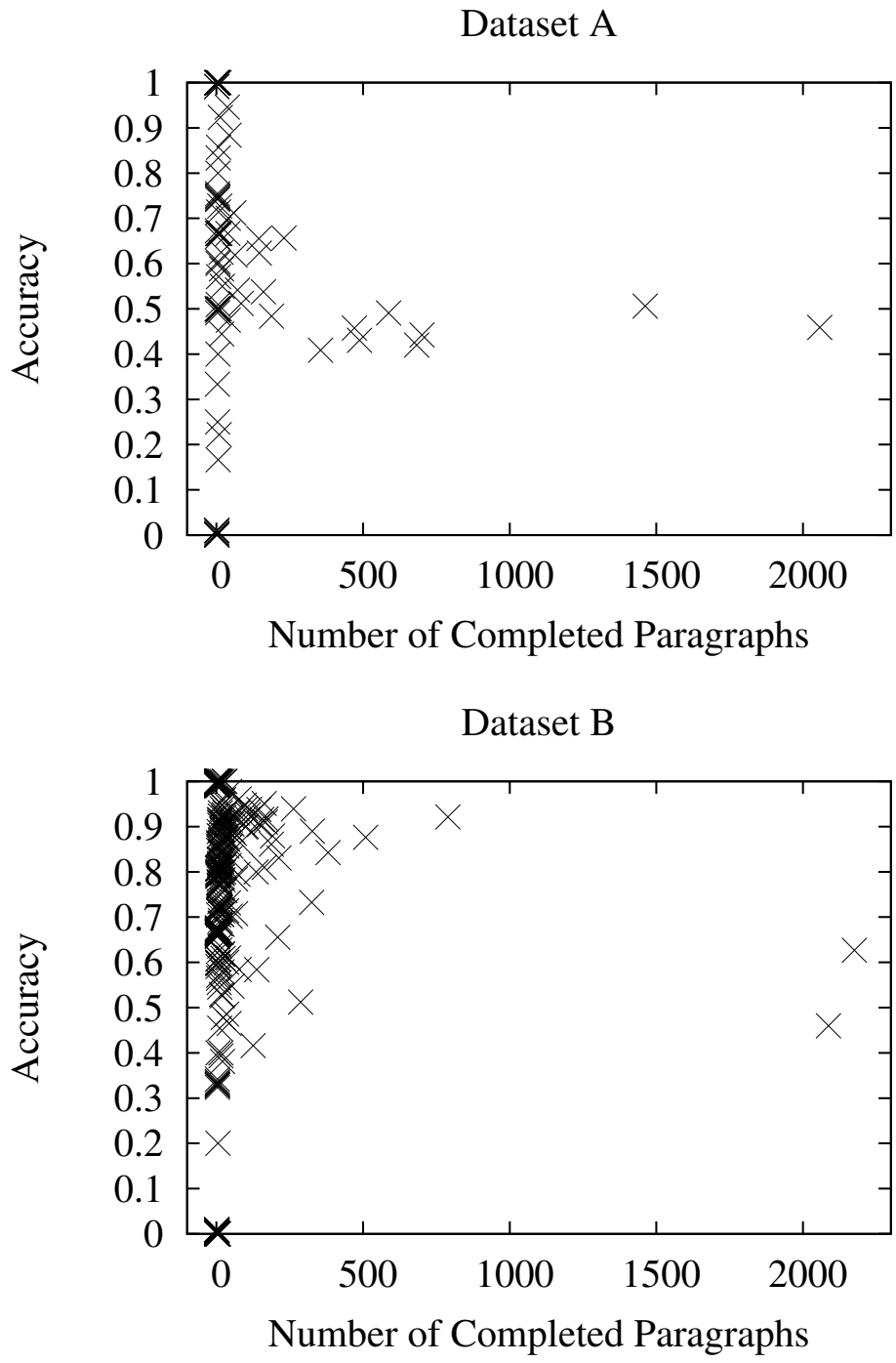

Figure 6.4.. Individual MTurker's accuracy vs. numbers of passages they completed. Each cross stands for an MTurker. Its abscissa value is total number of passages he completed. Its ordinate value is his annotation accuracy.

Dataset	A	B
Approval Rate	95	98
Workers per HIT	3	5
Duration (hrs)	9.55	47.63
Annotation Accuracy	49.37%	73.77%

Table 6.3.. Comparison of Datasets exported from AMT. The annotation accuracy improves with the increasing of the approval rate and workers per HIT.

# of Agreed Workers	# of HITs
Dataset A	
Two	1748 (61.20%)
Three	1108 (38.80%)
Dataset B	
Three	1068(37.39%)
Four	1030 (36.06%)
Five	758 (26.54%)

Table 6.4.. Inter Annotation Agreement for the 2856 question-passage pairs for both Dataset A and B.

nearly similar to results of random choice. We check the details of their judgements and find that they produce a large number of random labels in a short period. Regarding the necessary time of reading text and making decision, their behaviour in extremely short time is definitely impossible. Overall, the argumentation of HIT approval rates does not effectively filter more spammer workers as we expected. In practice, we rejected responses from noisy workers whose annotation accuracy scores below a threshold.

On the other hand, as the number of annotators per HIT increases, from three to five MTurkers the density of workers on the left top regions of figures, i.e. there was a significant positive correlation between the increment of workers with higher annotation accuracy and the increment of overall workers. On average MTurkers complete hundreds of assignments then terminate their participation as they probably lose their interests or persistence on completing the task. The improvement of annotation accuracy is a joint effect of increasing HIT approval rate and recruiting more people per HIT. In the HIT dashboard where online MTurkers can search and browse HITs, those tasks with more HITs are boosted to the top of retrieved results. MTurkers build message boards [7] to publish the most interesting or

[7] http://turkernation.com/

profitable HITs. HITs with large amount of assignments attract more workers, proportionally more spammer workers. To reduce the proportion of spammer workers, we adopt the principle of employing more workers per HIT as well as setting a higher HIT approval rate.

6.4. Related Work

Crowdsourcing makes it feasible to utilizing distributed human time and intelligence for solving problems that computers cannot yet deal with, such as those involved with basic conceptual intelligence and perceptual capabilities. AMT provides a virtual crowdsourcing venue to collect human intelligence and aggregate crowd wisdom. The advantages of low cost, huge workforce and flexible utility have attracted increasing research topics on utilizing and learning from crowdsourcing in NLP communities. We will discuss on contemporary work on application of crowdsourcing in this section. The purpose of this paper is twofold. It should serve as a self-contained introduction to this rapidly developing field of crowdsourcing application. We do not intend to thoroughly survey of all related work in various domains, however instead aim at classifying applications of crowdsourcing for NLP, IR and related fields, and give insights which may be helpful for the future work. We take the representative AMT as the execution platform.

The majority applications of crowdsourcing include resource collection for various tasks and human evaluation of system performance. The first systematic study of crowdsourcing for NLP was reported by Snow et al.[88]. They empirically examined five NLP tasks, including word sense disambiguation, word similarity, textual entailment, and temporal orderings of events. They proposed a Bayes-related technique to improve the annotation quality. Callison-Burch [10] provided in-depth analysis of crowdsourcing for complex data creation tasks, such as creating multiple reference translations and reading comprehension tests. Both papers highlighted the challenges and strategies for soliciting high-quality data from noisy online annotations. They concluded that non-expert labelers in aggregation can produce judgements highly agreeing with gold-standard experts, despite the behaviour of

individual MTurkers is less reliable.

In the spectrum of IR, a considerable amount of literature has been published on accomplishing relevance assessment with crowdsourcing. Kaisser et al. [46] investigated the impact of summary length on the quality of search results by conducting surveys via AMT. In their intensive analysis of user reports, they pointed out that search results best presented different lengths of summary snippets for different types of queries. Alonso et al. [2] evaluated the quality of search results produced by their time-based clustering algorithm combined with temporal snippets. Alonso and Mizzaro [3] performed five experiments on TREC retrieve data via AMT. Their comparisons with official TREC assessments suggested that crowdsourcing is a viable alternative for relevance assessment with offline human assessors.

The prior step of running experiments on AMT is to design and develop the task-oriented interface. The stereotypical crowdsourcing task is to let MTurkers read textual information and accordingly work on creating, collecting or assessing data. AMT provides alternative implementations of HTML/JS interface, Java or Flash applet to present annotation HITs with multimodal elements, such as images, sounds and videos. Sorokin and Forsyth [89] outsourced image annotation to AMT. They proposed four annotation modules and asked workers to identify objects in images by clicking on landmark points or drawing the boundary of a object from a picture. Chen and Dola [13] converted the task of writing paraphrases into describing the action in an Youtube video with a sentence. Their annotation framework could easily collect arbitrarily large training and test sets of independent linguistic descriptions of the same semantic content, which was vividly presented as a video. They designed an auxiliary crowdsourcing task on collecting eligible Youtube videos. Eventually they spent less than $5,000 on collecting 2,089 videos segments and 85,550 English descriptions over a two-month period. Novotney and Callison-Burch [76] transcribed conversational speeches in English, Korean, Hindi and Tamil by making MTurkers listening phone call records, and the MTuker-generated non-professional transcriptions is only 6% worse in quality than professional transcriptions with $1/30$ the cost. Experimental results

show that crowd-generated data was nearly as effective as gold standard data for training speech models.

Prior studies [100] have noted the importance of quality control for annotation, which is especially a critical issue for crowdsourcing annotation. The principle of quality assessment is to make sure users understand the task clearly, clean up occasional errors, detect and prevent cheating operations [89]. In their case study of image annotation, Sorokin and Forsyth [89] identified three representative strategies:

Multiply annotations. The basic and common strategy is to collect multiply annotations. The plurality of annotation makes it possible to select annotations in the ways of voting [46, 10], consensus [13] or averaging [3], although meanwhile the cost of annotation increases consequently.

Grading. An auxiliary grading task is set up so that another group of MTurkers provides gradings to help assess the quality of annotations. Callison [10] filtered the crowdsourcing translations by asking alternative group of MTurkers to judge those translations and abandon those with negative preferences.

Gold standard evaluation. A fraction of the annotating data is given with trusted annotations. The comparison of MTurker's annotation against the gold standard annotation can mostly reflect their overall performances.

Most MTurkers desire to make money from those micro works available on AMT [86], therefore it is important to retain a fair and pricing system to attract more MTurkers and keep high-quality workers for the consistent contribution. Chen and Dola [13] proposed a **two-tiered payment system**, which is involved with two successive groups of HITs. Those workers who perform well on the preliminary Tier-1 task are given access to Tier-2 tasks with much higher payment. This strategy provokes worker loyalty and requester reputation, and maintains a sustainable and stable higher-quality workforce retaining during the whole annotation duration.

The discussion so far is on posting independent and parallel crowdsourcing tasks, on the

other hand we can run crowdsourcing tasks iteratively. A series of tasks is run successively to recruit new MTrkers to alternate annotations and enhance the annotation generated by forerunning tasks. Turkit [8] is a task-managing software build on top of AMT, which supports simultaneously or iteratively submitting and publishing tasks. TurKit was used to improve some artifacts, such as handwriting recognition (*Human OCR*). It posts a HIT for recognizing words from a image, then show the recognized words and images to another worker, who works on improving the recognition. The advantage of iterative task over the parallel task is that those iteration can be cycled and resubmitted any times without increasing the cost. Results in Little et al. [58] showed that the imperative programming paradigm effectively uses AMT workers as subroutines to gradually produce higher quality data.

Some experiments, such as the training of speech recognition [76], suggested from the perspective of improving performance of real systems, it is better to collect more data than to improve online annotation quality, however quality control is still the cornerstone for effective crowdsourcing experiments in general cases.

Previous efforts at QA corpus construction focus on soliciting more precise annotated data. Kaisser et al. [47] constructed a corpus of question-sentence pairs for the TREC factoid question and employed experts to further cleaned the corpus and tagged how sufficiently a sentence supports its question. To create a *why* QA corpus, Morzinski et al. [73] adopt a dive-and-conquer design strategy to split the creation of a why QA corpus into three subtasks. The first HIT is on creating questions, which asked MTurkers to write a why question on a selected Wikipedia article. Secondly second group of MTukers is required to select sentences from the original articles for answering the created questions. In the final task workers paraphrased each question to provide variation of questions. The combination of those results thereby is taken as the Why QA collection.

Crowdsourcing has attracted great attention from the academic research community on NLP. There are many successful studies, such as generating cross-lingual textual

[8] groups.csail.mit.edu/uid/turkit/

entailment [74], speech recognition [65, 66, 63], active learning for NLP [52], named entity recognition [52] and so on. Crowdsourcing contributes to lots of applications in many fields including creating large-scale image annotation database ImageNet [26], video database LabelMe [82] and studying human computer interaction [9]. Many open NLP and IR evaluations such as TREC Blog track 2010 [64] and several tasks at CLEF2011 [10] have used the crowsourcing techniques for system evaluation. The evidence from these studies and trends suggests that crowdsourcing is a reliable alternative to traditional methods on data creation and assessment. Besides AMT, we have many websites offering diverse infrastructures for different tasks, such as Crowdflower [11], Livework [12], Elance [13], Kaggle [14] and so on.

6.5. Conclusion

In this chapter we introduced a novel crowdsourcing methods for various data assessing, creation and generation NLP and IR tasks via Amazon Mechanical Turk. Firstly we briefly describe the function and workflow of planning, designing and submitting Human Intellingence Tasks via AMT. Then we described the details of our crowdsourcing experimental procedures on judging relevant passages for TREC list questions, and discussed the methods we used to control the quality of data generated by anonymous online workers. Finally we reviewed related research trends featured with crowdsourcing judgement and annotation, and concluded the key aspects of designing and managing crowdsourcing tasks. We explained several strategies for the quality control of crowdsourcing results. Many questions on quality control are open to further investigations. In the next chapters we will introduce more effective methods on modelling the annotation process to increase the annotation quality with supervised and unsupervised learning methods.

[9] crowdresearch.org/chi2011-workshop/
[10] http://clef2011.org/
[11] http://crowdflower.com/
[12] http://www.livework.co.uk/
[13] https://www.elance.com/
[14] http://www.kaggle.com/

7. True Annotation Learning

7.1. Introduction

Due to the variation of individual worker's reliability and each HIT's complexity, crowd annotations are not perfect (see Table 6.4 and Figure 6.4). How to optimally combine annotations from multiple labelers and learn the true label is of great significance to automatic data annotation. Hereby we compare three approaches to this task: supervised Naive Bayesian model [88], unsupervised GLAD model [108], and the Majority Voting (MV) as the baseline. The estimate of true label will be used to learn and evaluate the passage ranking methods in Chapter 8.

7.2. Majority Voting

With the principle that the majority rules, the majority voting method assumes all workers exhibit identical expertise and therefore have equal votes. We choose the labels on which the majority of them agrees as an estimate of the actual gold standard. Our supporting passage judgement is a binary classification problems, so we simply use the majority label as the true labels, i.e.,

$$\hat{y_i} = \begin{cases} 1 & \text{if } \frac{1}{R}\sum_{j=1}^{R} y_i^j > 0.5 \\ \\ 0 & \text{if } \frac{1}{R}\sum_{j=1}^{R} y_i^j < 0.5 \end{cases} \tag{7.1}$$

The problem with this equal voting assumption is that it fails to take the variance of *labeler expertise* and *annotation difficulty* into account. In online annotation scenarios, if the majority are noisy or adversarial workers who give the same incorrect labels to an annotating instance, the majority voting would favour major incorrect labels and ignore true labels in the minority. The following two methods are more effective at learning true annotation owning to modelling the diversity of tasks and labelers.

7.3. Naive Bayes

Snow et al. [88] introduced a multinomial Naive-Bayes-Type (NBT) model to estimate the worker's expertise and weight each worker's vote with their performance likelihood.

They consider the annotation learning as a supervised learning problem which approximates the target function $P(x_i|\mathbf{l}_i)$. For each passage i, we assume x_i is a boolean-valued label variable. $x_i = 1$ means the passage supports the question, and $x_i = 0$ vice versa. $\mathbf{l}_i$ is a vector containing n boolean labels. In other words, $\mathbf{l}_i = \{l_{ij} : j = 1, \ldots, J\}$, where l_{ij} is the label given by a worker j to a passage i.

The conditional probability of a passage's true label x_i given its annotations $\mathbf{l}_i$ is calculated to determine the true label. Using the Bayes rule,

$$P(x_i|\mathbf{l}_i) = \frac{P(\mathbf{l}_i|x_i)P(x_i)}{P(\mathbf{l}_i)} \tag{7.2}$$

where each worker's labels are assumed as conditionally independent of each other given the true label x_i, then we have

$$P(\mathbf{l}_i|x_i) = \prod_j P(l_{ij}|x_i) \tag{7.3}$$

We rewrite the Equation 7.2 as

$$P(x_i|\mathbf{l}_i) = \frac{P(x_i)\prod_j P(l_{ij}|x_i)}{P(\mathbf{l}_i)} \tag{7.4}$$

while can be simplified to the following as the denominator does not depend on x_i.

$$P(x_i|\mathbf{l}_i) \propto P(x_i)\prod_j P(l_{ij}|x_i) \tag{7.5}$$

$P(x_i|\mathbf{l}_i)$ relies on the estimation of the performance likelihood $P(l_{ij}|x_i)$ of each worker's label and prior probability $p(x_i)$. Both of them are estimated, respectively, as the relative occurrence frequencies in the training set. Finally we can then use these estimates together with the Bayes rule above to determine $P(x|\mathbf{l}_k)$ for any new instance $\mathbf{l}_k$.

We can estimate these parameters using maximum likelihood estimates. $p(x_i)$ is defined over the counting of labels. The estimation of each worker's performance likelihood $P(l_j|x)$ is derived from incorporating their annotation accuracy with regard to true labels of passages they completed, e.g., $P(l_j = w|x = t)(w, t \in \{0, 1\})$ measures the ratio of the worker j's labels are class w given truth labels are class t, and is fit with Laplace smoothing.[1]

$$\begin{aligned} &P(l_j = w|x = t) \\ &= \frac{\sum_{k\in\Phi_j}\delta(l_{kj} = w \wedge x_k = t) + 1}{\sum_{k\in\Phi_j}\delta(x_k = t) + |\Phi||S|} \end{aligned} \tag{7.6}$$

where, Φ_j denotes the set of passages worker j completed. Φ is the complete set of all passages. $|S|$ is the number of assignments per HIT.

Given all workers' performance likelihood for passage i, the true label x_i is judged using the posterior log odds:

$$Q(R) = \log\frac{P(x_i = 1|\mathbf{l}_i)}{P(x_i = 0|\mathbf{l}_i)}$$

[1] $\delta(x)$ is 1 if its logical argument x is true and 0 otherwise.

$$= \sum_j \log \frac{P(l_{ij}|x_i = 1)}{P(l_{ij}|x_i = 0)} + \log \frac{P(x_i = 1)}{P(x_i = 0)} \tag{7.7}$$

If the log odds $Q(R)$ is positive, the label of a passage is class 1, which means the passage supports the answer.

7.4. GLAD Model

The Naive Bayes model only focuses on modeling the expertise of labelers, whereas in realistic annotation scenarios, some annotation instances are harder to tackle than others. Additionally it is impossible to measure the expertise of labelers when the gold standard is not available. The Generative model of Labels, Abilities and Difficulties (GLAD) [108] infers the true label by capturing task difficulty and the labeler expertise in an unsupervised Exception-Maximization (EM) model.

Figure 7.1 depicts the causal structure of the GLAD model. We do not known true labels X_i, labeler accuracy values α_j and task difficulty values β_i, therefore we assume that all those variables are sampled from an known prior distribution, which determines the observed labels according to Equation 7.8. Given a set of observed labels $\mathbf{l}$ contributed by MTurkers, we aim at inferring simultaneously the most likely values of $\mathbf{X} = X_j$ (the true labels) as well as the labeler accuracies $\alpha = \alpha_j$ and the task difficulty parameters $\beta = \beta_i$. The inference process is based on the EM algorithm.

The difficulty of passage i is modelled using the variable $\beta_i \in [0, \infty)$ where $\beta_i > 0$. Here $\beta_i = \infty$ means the passage is very hard to judge. $\beta_i = 0$ means the passage is so easy that most workers always judge correctly.

The expertise of a worker j is modelled by the variable $\alpha_j \in (-\infty, +\infty)$. Here an $\alpha_j = +\infty$ means the worker always makes correct labels, whereas $\alpha_j = -\infty$ means the worker always judges incorrectly. Then for worker j for passage i, the posterior probability for

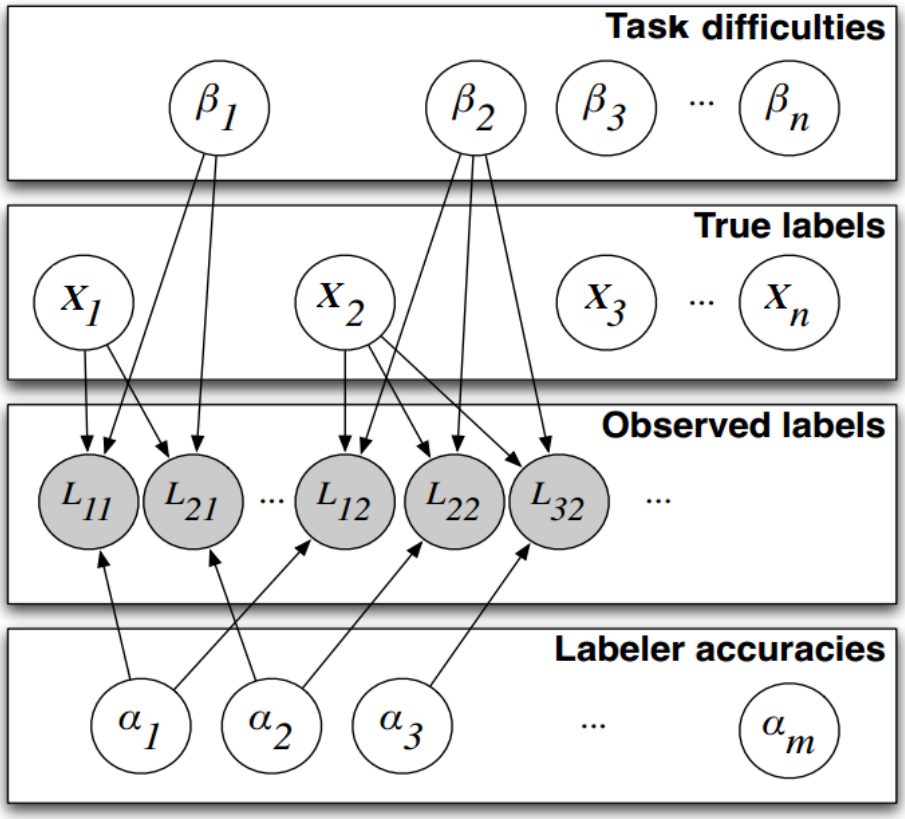

Figure 7.1.. Causal structure of GLAD model for inferring the hidden variables including task difficulties β, true label X, and labeler accuracies α given the observed labels L. Only the shaded variables are observed.

$x_i = 1$ is defined as,

$$P(l_{ij} = x_i|\alpha_j, \beta_i) = \frac{1}{1 + e^{-\alpha_j/\beta_i}} \tag{7.8}$$

The EM algorithm is designed as an efficient iterative algorithm for estimating the maximum likelihood (ML) in the presence of hidden data. It obtains maximum likelihood estimates of true labels $\mathbf{X}$ and parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ given the observed data.

Each iteration of the EM algorithm consists of an Expectation(E) – step and a Maximization(M) – step.

E step: The posterior probabilities of all $x_i \in \{0, 1\}$ given the estimate of parameters $\boldsymbol{\alpha}, \boldsymbol{\beta}$ from last M step and the worker labels:

$$\begin{aligned} P(x_i|\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}) &= P(x_i|\mathbf{l}_i, \boldsymbol{\alpha}, \beta_i) \\ &\propto P(x_i|\boldsymbol{\alpha}, \beta_i) P(\mathbf{l}_i|x_i, \boldsymbol{\alpha}, \beta_i) \\ &\propto P(x_i) \prod_j P(l_{ij}|x_i, \alpha_j, \beta_i) \end{aligned} \tag{7.9}$$

where the conditional independence assumption from the graphical model leads to $P(x_i|\boldsymbol{\alpha}, \beta_i) = P(x_i)$.

M step: To maximize the standard auxiliary function Q, which is defined as the expectation of the joint log-likelihood of the observed and hidden variables $(\mathbf{l}, \mathbf{X})$ given the parameters $(\boldsymbol{\alpha}, \boldsymbol{\beta})$ estimated during the last E-step:

$$\begin{aligned} Q(\boldsymbol{\alpha}, \boldsymbol{\beta}) &= E[\ln p(\boldsymbol{l}, \boldsymbol{x}|\boldsymbol{\alpha}, \boldsymbol{\beta})] \\ &= E\left[\ln \prod_i \left(p(x_i) \prod_j p(l_{ij}|x_i, \alpha_j, \beta_i) \right) \right] \\ &= \sum_i E[\ln P(x_i)] + \sum_{ij} E[\ln P(l_{ij}|x_i, \alpha_j, \beta_i)] \end{aligned} \tag{7.10}$$

Gradient ascent algorithm is employed to find values of α, β that locally maximized Q.

We apply different initialization on dataset A and B introduced in Table 6.3. For the dataset A, as a large proportion of labels are judged incorrectly, α need be made very low for $\alpha > 0$. We used Gaussian priors $(\mu = 0.0001, \sigma = 0.0001)$ as priors for α and initialized all $\mathbf{X}$ with 0.0001. For dataset B, optimal priors α values are Gaussian priors $(\mu = 0.9, \sigma = 0.9)$ the $\mathbf{X}$ are initialized with 0.5. We re-parameterized $\beta = e^{\beta'}$ and imposed a Gaussian prior $(\mu = 0.0001, \sigma = 0.0001)$ on β' for dataset A and $(\mu = 0.9, \sigma = 0.9)$ for dataset B. The label of a passage is class 1 when $P(x_i = 1|\mathbf{l}, \boldsymbol{\alpha}, \boldsymbol{\beta}) > 0.5$.

7.5. Results

To compare the effectiveness of learning methods, the gold-standard annotations are used as ground truth judgements. We measured the effectiveness in term of proportion of correctly inferred labels. Table 7.1 showed the accuracy of each approach against two different levels of annotation accuracies. The NBT model is trained and tested via 20-fold cross validation on the whole dataset. The application of both methods brings significant accuracy improvements over the baseline in learning true annotations. Contrary to results presented in [108], the NBT model makes fewer errors than the GLAD model. The probable reason is as following: NBT method makes use of pre-labeled ground truth labels; Although Whitehill et al. [108] claimed the GLAD's advantage of modeling task difficulty might be very important, experimental results with different values of β rarely changed in our case, therefore GLAD's performance is somehow weakened by unsuccessfully modeling the passage difficulty.

In order to explore the influence of HIT approval rate on the performance of learning methods, we perform a simple simulation: for each passage in dataset B, 3 labels are randomly chosen from 5 labels (named as Dataset B_{3W}), on which we test those three approaches. The simulation is repeated 100 times to smooth out variability between trials. The average accuracy is shown in Table 7.1. Comparison between dataset A and B_{3W}

	A	**B**	$\mathbf{C_{3W}}$	$\mathbf{B_{3W}}$
AA	49.37%	73.77%	63.63%	72.02%
MV	49.61%	82.98%	67.09%	79.06%
GLAD	54.52%	89.81%	67.51%	85.04%
NBT	61.75%	91.36%	81.79%	87.47%

Table 7.1.. Accuracies of the approaches on dataset A and B with different annotation accuracies (AA).

indicates that improving HIT approval rate results in better AMT online annotation accuracy and therefore leads to significant performance improvements. From dataset B_{3W} to B, we can see that recruiting more labelers per HIT can also obviously boost performance.

We merge dataset A and B into dataset C (8 labelers per HIT and annotation accuracy 63.58%), on which we further investigate the effect of varying the number of labelers per HIT. Figure 7.2 demonstrates the analytical relationship between the accuracy of estimated labels and the number of labelers, for different approaches. As expected the performance of NBT and the majority voting model improves with larger numbers of labelers, while the GLAD model shows unstable performance and does not show advantage over the majority voting method. Dataset A, $\mathbf{C_{3W}}$ and $\mathbf{B_{3W}}$ all employ 3 labelers per HIT. Comparisons of their results in Table 7.1 indicate that as the annotation accuracy increases steadily in those three datasets, the performance of all methods increases. The GLAD model works noticeably better on dataset with better quality (e.g. dataset B and $\mathbf{B_{3W}}$). When the annotation accuracy is low (49.37% of dataset A), all methods tend to show low accuracy due to the influence of large amount of noisy and adversarial labels.

Our results strongly suggest setting higher HIT approval rate (normally 98%) for the practice with AMT assures higher online annotation accuracy, and so those three approaches can recover the true labels more accurately. Additionally NBT method mainly relies on the workers' performance likelihood estimated on the training data. If a number of new workers appear only in the testing data, their performance can not be estimated during the training stage, while the GLAD model does not suffer from this new worker problem. When ground truth labels are not available and AMT annotations show reasonable accuracy,

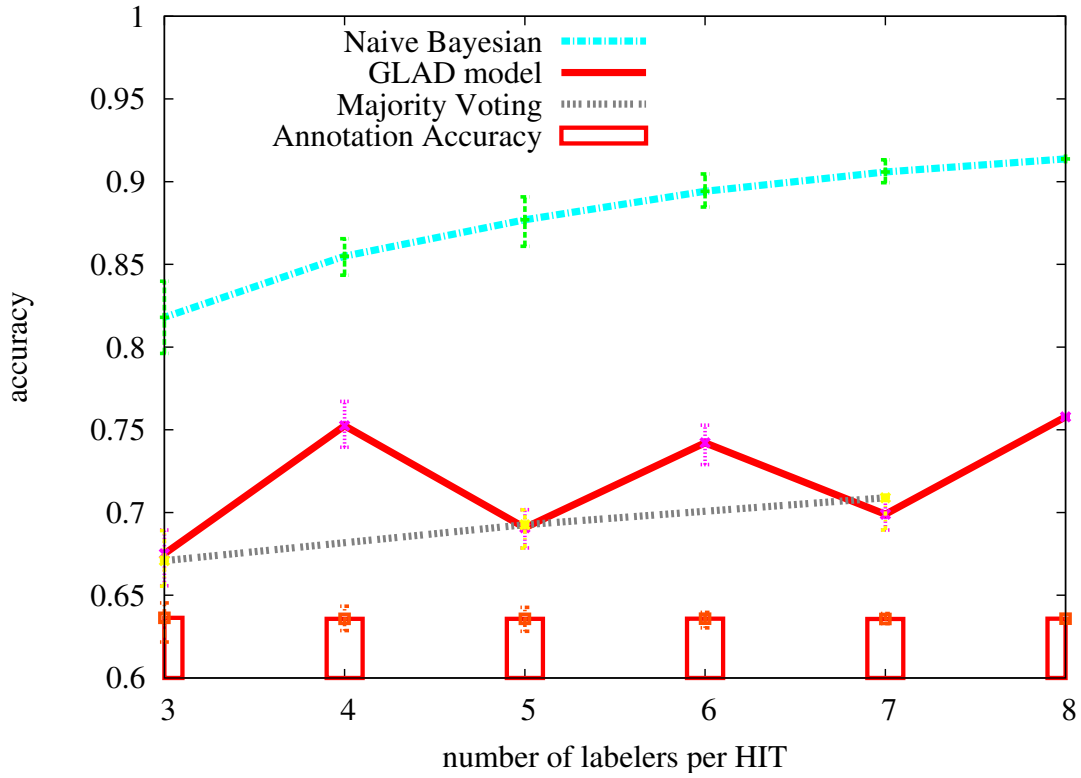

Figure 7.2.. Accuracies of the approaches on dataset C vs. number of labels per HIT. All experimental trials are performed over 100 random samplings of labelers for all passages. The majority voting only consider odd numbers of labelers. For GLAD model, We used Gaussian priors $(\mu = 0.0001, \sigma = 0.0001)$ for α, Gaussian priors $(\mu = 0.0001, \sigma = 0.0001)$ for β' and the $\mathbf{X}$ are initialized with 0.0001.

GLAD is more advantageous than the NBT model since it can learn true labels without supervision.

7.6. Conclusion

In Chapter 6 we constructed a new corpus of supporting passage collections for list question in TREC QA 2004. In this chapter to improve the annotation quality, we compared three approaches on selecting accurate annotations in AMT results, and investigated the influence of mislabeled data and a number of labelers per HIT on their performance. Experiments show that with careful design of tasks and appropriate approaches to select true labels, high-quality labels can be automatically learned from AMT non-expert annotations. We also suggested that better online annotation quality leads to better performance of learning methods. In the next chapter, we will use the learned annotation to train the statistical passage ranking methods.

The question-passage corpus and AMT results are available via our group's homepage [2].

[2] http://www.lsv.uni-saarland.de/personalPages/fangxu/data/listq.tar

45.3 APW19980615.1543.8 South Korea
A third investment involves the Korea Trade Enhancement Facility (KTEF), a US $100 million trade enhancement facility established by IFC with Sumitomo Bank Ltd to expand trade finance to **South Korea**.
45.3 XIE19990902.0037.1 Colombia
IFC's investment will finance the first stage of development of the Bolivar Block in **Colombia**'s Middle Magdalena Valley. This phase will include drilling nine wells and constructing facilities and transmission pipelines to produce up to 30,000 barrels of oil per day which will be exported via Covenas on the country's Caribbean coast.
45.3 XIE19980112.0166.1 Kenya
More than 66 million Dollars have been committed by IFC, the private sector lending arm of the World Bank, to projects in **Kenya** since 1970, the East African weekly reported today .
45.3 XIE19960126.0179.2 Pakistan
Addressing a meeting at the Lahore Chamber of Commerce and Industry, he said that the IFC would continue its financial assistance in **Pakistan**'s investment activities by further expanding its operation.
45.3 XIE19980112.0166.0 ''Kenya'', ''Uganda'', ''Tanzania''
NAIROBI, January 12 (Xinhua) -- More and more private sector projects in **Kenya**, **Uganda** and **Tanzania**, all the three members of the East Africa Cooperation (EAC), have been getting funding from the International Finance Corporation (IFC) over recent years.
45.3 XIE19970626.0057.4 Mozambique
The IFC is a member of the World Bank Group, and the largest multilateral source of equity and loan financing for private sector projects in developing countries. Up to date, the IFC has invested over 11 million dollars for six projects in **Mozambique**.
45.3 XIE19961024.0231.0 Philippines
WASHINGTON , October 23 (Xinhua) -- The International Finance Corporation (IFC) today announced the approval of 37.5 million U.S. dollars in loan and equity to finance a shipping company in the **Philippines**.
45.3 XIE19960523.0173.0 Indonesia
WASHINGTON , May 22 (Xinhua) -- The International Finance Corporation (IFC) has agreed to provide up to112.35 million U.S. dollars for an expansion of a ceramic roof tile manufacture project in **Indonesia**.

Table 7.2.. Examples of the *question ID*, *passage ID* and *answer string* following with the *supporting passage* for the question "What countries has the IFC financed projects in?"

8. Learning to Rank Supporting Passages for List Questions

8.1. Introduction

Passage retrieval methods are widely used by most QA systems in TREC as the middle layer between document retrieval and answer extraction. Appropriate passage retrieval methods can largely reduce the search space for answer extraction, from a complete document to precise and compact text excerpts.

To tackle the problem of passage retrieval, many statistical ranking models have been proposed in IR literature. The objective of passage retrieval for QA is to find passages that are not only relevant to one question but also sufficiently support their containing answers to the question. Previous studies focused on solving passage retrieval for factoid questions. In this thesis we also focus on list questions, which require a list of concise, succinct and distinct answers for each natural language question, e.g., the question "*What countries have IFC financed projects in?*" has 42 distinct answers [1].

Given the large amount of training data, recent work has shown the feasibility of building effective ranking models for passage ranking tasks. Learning to rank techniques have been successively applied to passage retrieval for the questions on a single fact, e.g., manner questions [91], *why* questions [99] and quantity consensus questions [7]. Rather than manually tuning parameters on a few features in traditional retrieval models, learning

[1] TREC 2004 answer sets are available at http://trec.nist.gov/data/qa/t2004_qadata.html.

to rank methods can incorporate a wide range of relevance features, thus they outperform traditional passage retrieval methods. The flexibility and scalability make it possible to apply learning to rank techniques to different domains.

In this chapter, we propose learning to rank techniques to rank supporting passages for list questions. We discuss learning to rank techniques under the framework of Support Vector Machine (SVM) with 3 different categories – pointwise Support Vector Regression (SVR), pairwise ranking SVMs, and listwise SVM^{map}. Experimental results show that learning to rank techniques with only 12 features significantly outperform traditional retrieval models.

The rest of this chapter is organized as follows. In Section 8.2, we present our feature design for measuring relevance between question and passages. In Section 8.3, we describe three representative SVM-based models for learning to rank. In Section 8.4 we present our experimental evaluation and analysis. Finally, in Section 8.5 we conclude the chapter.

8.2. Features

In this section, we will investigate different features on estimating relatedness between questions and answers. We first introduce traditional ranking features on measuring occurrence of question words in the passage. To catch distant relations, we utilize lexical proximity between question word and named entities (NEs) in passages. Some features can incorporate semantic similarities of words between questions and passages. We also define several features based on web-retrieved information.

8.2.1. Question Representation

Surdeanu et al. [91] introduced features to represent lexical, syntactic and semantic information for questions, however the usage of linguistic features yields only a small performance increase. The output of linguistic processing contains errors, and most

syntactic and semantic processing tools can only resolve intra-sentential relations. To avoid those negative affects, we adopt BOW representation, as well as question types and question series phrases as topic representations.

Question serial phrase: The TREC QA dataset groups questions into different series of targets (topics). Most topics are NEs, e.g. *series 88: United Parcel Service* and *series 72: Bollywood* [2]. We used the question topics to construct features in two ways: The question topic phrase supplement abbreviations, pronouns and nouns in the question, which are considered as query expansions, such as *UPS* for *United Parcel Service*; each question is about an attribute on the question topic, e.g., all five questions in the *series 88* ask about various attributes of the company UPS. We measure the association of question topics and NEs in the passages.

Question type is the anticipated type of answers for a question such as questions in *topic 88*:

- **Location**: "Where is the American Enterprise Institute located?"
- **Person**: "Who is the CEO of UPS?"
- **Date**: "When was UPS's first public stock offering?"

We have adopted three general question classes, namely Person, Location and Organization, as they occur most frequently in TREC QA dataset.

8.2.2. BOW Ranking Features

Many ranking models have been proposed in IR literature for document retrieval [62]. The traditional models measure document relevance based on the occurrences of query terms in the document. Term frequencies (TFs) and inverse document frequencies (IDFs) have been widely used as effective representations of queries and documents. The TF of a term is simply the count of its occurrences within a document normalized by the length

[2] http://trec.nist.gov/data/qa/2005_qadata/QA2005_testset.xml

of the document, and the IDF reflects the commonness of a term across all documents, which is defined as

$$\text{idf}(w) = \log \frac{|D|}{n(w)}, \tag{8.1}$$

where $|D|$ is the total number of documents in the collection, $n(w)$ is the number of documents where w appears. [3]

We treat the question terms as queries and passages as documents to retrieve. Each passage is determined by occurrences of question words in two fields F: the current passage, four preceding and following contextual passages. For each field F, three features are calculated:

$$\textbf{TFSUM}: \sum_{w \in q \cap F} tf(w, F), \tag{8.2}$$

$$\textbf{IDFSUM}: \sum_{w \in q \cap F} idf(w), \tag{8.3}$$

$$\textbf{TFIDFSUM}: \sum_{w \in q \cap F} tfidf(w, F) = \sum_{w \in q \cap F} tf(w, F) \cdot idf(w), \tag{8.4}$$

$idf(w)$ is calculated in the collection of all passages for all questions.

The BOW representation is limited by the strong assumption of word independence. More sophisticated features are thus developed to discover sufficiently relatedness between questions and passages.

8.2.3. Query Proximity Features

Previous works on answer ranking [7] and question classification [39] considered high-order N-grams as units to represent texts. It is an indirect way to capture the word proximity. However, query tokens may not appear far away from each other. We present new proximity features to reflect the dependency of non-adjacent query tokens.

[3] Several variants of TF and IDF have been suggested; see Manning, Raghavan and schütze [62].

The positional term frequency proposed in [60] is used to capture the dependency between non-adjacent question terms by considering the pairwise proximity.

$$\mathbf{ProxTF} = \sum_{w \in q} \sum_{j=1}^{N} c(w,i)k(i,j) \tag{8.5}$$

$c(w,i)$ is the count of term w at the position i in the passage. $k(i,j)$ is the propagated count to position i from a term at position j, which serves as a discounting factor and non-increasing function of $|i-j|$, i.e., $k(i,j)$ gives more weight to the token closer to position j. $k(i,j)$ is defined as the Gaussian kernel,

$$k(i,j) = \exp\left[\frac{-(i-j)^2}{2\sigma^2}\right], \tag{8.6}$$

where the parameter σ controls the propagation range of each term. We set σ to be the length of passage N to spread the kernel curve over a passage.

8.2.4. NE Proximity Features

We use the proximity term frequencies to estimate the dependency between query terms and NEs. Intuitively, passages with more question-related NEs are more likely to be relevant, so we consider all NEs matching the expected question type as answer seeds and estimate the relevance likelihood by the positional term frequency. We introduce the proximity between NE $\mathbf{seed}_i$ and a question token using the Gaussian kernel.

We use four proximity features for passage ranking:

- maximum proximity of $\mathbf{seed}_i$ to any question token, which favours query terms nearest to a answer seed;
- proximity of $\mathbf{seed}_i$ to the rarest question token (with biggest IDF), which focuses on specific query terms;
- IDF-weighted average of proximity to all question tokens. Each individual proximity

is weighted by its specificity — IDF.

- proximity of $\mathbf{seed}_i$ to the smallest IDF question token, introduced to increase the diversity of features.

When calculating the Gaussian kernel for a question "*list universities that she visited.*", if a answer seed contains a question term such as "*university*", the distance $\|i - j\|$ of term "university" is defined as zero as it is a sub-word of the seed.

8.2.5. WordNet-based Features

Those aforementioned features are oblivious of the meanings of words in QA contexts. Mihalcea et al. [69] derived a text-to-text relatedness from combining word-to-word semantic similarity. The objective is to incorporate semantic similarities of words to support semantic matching between questions and passages. They calculated the text similarity based on the *similarity* and *specificity* of words.

The specificity of a word is determined using IDF (Equation 8.1), so higher weights are given to a match between a pair of specific words (e.g. *collie* and *sheepdog*), rather than similarity measured between generic concepts (e.g. *get* and *become*). For each question token w, we try to identify the token in the passage that has the highest semantic similarity $maxSim(w, p)$. The similarity between q and p is therefore determined as follows:

$$sim(\boldsymbol{q}, \boldsymbol{p}) = \frac{\sum_{w \in \boldsymbol{q}} maxSim(w, \boldsymbol{p}) \cdot idf(w)}{\sum_{w \in \boldsymbol{q}} idf(w)} + \frac{\sum_{w \in \boldsymbol{p}} maxSim(w, \boldsymbol{q}) \cdot idf(w)}{\sum_{w \in \boldsymbol{p}} idf(w)} \tag{8.7}$$

The semantic similarity is calculated between words with the same POS. We make use of the structural information embedded in the hierarchical structure of WordNet [4] to evaluate the semantic similarity between concepts. There are several similarity metrics based on edges and nodes in WordNet semantic hierarchy, namely the conceptual distance metric and information content metric respectively.

[4] wordnet.princeton.edu

Path Similarity is the inverse of the shortest node-counting path $length(c_1, c_2)$ between corresponding WordNet synsets c_1 and c_2 for two lexicalized concepts.

$$sim_{path} = \frac{1}{length(c_1, c_2)} \tag{8.8}$$

Leacock & Chodorow Similarity [53] also relies $length(c_1, c_2)$ on measuring the semantic relatedness.

$$sim_{lch}(c_1, c_2) = -\log \frac{length(c_1, c_2)}{2D_{WN}} \tag{8.9}$$

where D_{WN} is the maximum depth of the taxonomic hierarchy of WordNet.

Wu-Palmer Similarity [113] measures the similarity of two concepts by how closely they are related in the hierarchy of WordNet within the domain of their Least Common Subsumer $LCS(c_1, c_2)$, which is defined as the ancestor node common to c_1 and c_2 whose shortest path to the root node is the longest.

$$sim_{wup}(c_1, c_2) = \frac{2 \cdot depth(LCS(c_1, c_2))}{depth(c_1) + depth(c_2)}. \tag{8.10}$$

Resnik Similarity [81] defines the Information Content,

$$IC = -log\ p(c) \tag{8.11}$$

of their LCS in WordNet. $p(c)$ is the occurrence of instances of concept c in a large corpus, such as Brown corpus [5]. The pairwise similarity is estimated by

$$sim_{res}(c_1, c_2) = IC(LCS(c_1, c_2)). \tag{8.12}$$

Jiang-Conrath Similarity [44] is a hybrid metric combining the distance-derived similarity

[5] Brown corpus, from the ICAME Collection of English Language Corpora Second Edition, 1999, http://www.hit.uib.no/icame/cd

and the information content of their LCS.

$$sim_{jcn}(c_1, c_2) = \frac{1}{IC(c_1) + IC(c_2) - 2 \cdot IC(LCS(c_1, c_2))} \tag{8.13}$$

Lin Similarity models the pairwise similarity by the commonality (estimated by the numerator in Equation 8.14) and difference (denominator in Equation 8.14) between two concepts [57].

$$sim_{lin}(c_1, c_2) = \frac{2 \cdot IC(LCS(c_1, c_2))}{IC(c_1) + IC(c_2)} \tag{8.14}$$

8.2.6. Web Popularity-based Features

The previous features assess the relevance of a passage by matching surface string or semantic taxonomy structure, while we can identify the degree of similarity between words based on models of distributional similarity derived from large text collections.

One natural way to access a large data collection is to search web pages with the help of commercial search engines such as Google. Those search engines automatically expand tokens in various morphological forms and exhaustively retrieve a large mount of web documents. Statistics and information derived from web corpora can be used to estimate relatedness between texts. Most relatedness measures take advantage of the number of hits returned by searching a term.

Here we represent a question with its included NEs, therefore we defined the pairwise sentential similarity as the sum of web-based similarity between all NEs $\boldsymbol{t_q}$ from a question sentence and all NEs $\boldsymbol{t_a}$ in the passage. Given two NEs t_q and t_a, we define the following metrics.

Pointwise Mutual Information (PMI) is based on word co-occurrence calculated from large-scale corpora, defined as,

$$PMI(t_q, t_a) = \log \frac{p(t_q + t_a)}{p(t_q)p(t_a)}, \tag{8.15}$$

$t_q + t_a$ denotes the union of t_q and t_a as one query. $p(w)$ is approximated as $f(w)/M$. M is the number of web pages indexed by Google. $f(w)$ is the number of hits for the term w. $f(t_q + t_a)$ results from submitting t_q and t_a together to a search engine. We obtain the following PMI_{WEB} measure:

$$PMI_{WEB}(t_q, t_a) = \log \frac{M \cdot f(t_q + t_a)}{f(t_q) f(t_a)} \tag{8.16}$$

Corrected Conditional Probability (CCP) [61] measures the relatedness between an NE pair by

$$CCP(t_q, t_a) = \frac{p(t_q + t_a)}{p(t_q) p(t_a)^{\frac{2}{3}}} = \frac{M^{\frac{2}{3}} f(t_q + t_a)}{f(t_q) f(t_a)^{\frac{2}{3}}} \tag{8.17}$$

$f(t_a)^{\frac{2}{3}}$ is used to avoid the influence of high-frequency words and patterns.

Normalized Google Distance (NGD) [18] is a dissimilarity measures that can be obtained as follows,

$$NGD(t_q, t_a) = \frac{G(t_q, t_a) - \min(G(t_q), G(t_a))}{\max(G(t_q), G(t_a))} \tag{8.18}$$

where $G(x, y)$ is the "Google code" function, $G(x, y) = -log f(x, y)$. The Google code applies Information Content in Equation 8.11 on web corpora. If search terms occur more frequently together on the same web pages than on separate pages, they are intended to be more similar. The normalized Google distance is thus given as

$$\frac{\max\{\log f(t_q), \log f(t_a)\} - log f(t_q + t_a)}{log M - \min\{\log f(t_a), \log f(t_q)\}} \tag{8.19}$$

8.2.7. Web-based Kernel Function

We want to use web search results as the context feature for a term and measure the similarity between terms by leveraging these features. We adopted the fine-grained web-based kernel function [83], which measures the similarity by capturing the semantic

contexts of top ranked search snippets instead of simply measuring their popularity. We first formalize the kernel function for calculating semantic similarity. Let x represents a short text snippet. The *query expansion* of x, denoted $QE(x)$, is computed by the following steps:

1. Submit x as a query to a search engine S.
2. Let $R(x)$ be the set of (at most) n retrieved documents $d_1, d_2, \ldots, d_n$.
3. Compute the TFIDF term vector v_i for each document $d_i \in R(x)$.
4. Truncate each vector v_i to include its m highest weighted terms.
5. Let $C(x)$ be the centroid of the L^2 normalized vectors v_i:

$$C(x) = \frac{1}{n} \sum_{i=1}^{n} \frac{v_i}{\|v_i\|_2} \tag{8.20}$$

6. $QE(x)$ is defined as the L^2 normalization of the centroid $C(x)$:

$$QE(x) = \frac{C(x)}{\|C(x)\|_2}. \tag{8.21}$$

The semantic similarity kernel between two entities x and y is defined as

$$K(x, y) = QE(x) \cdot QE(y) \tag{8.22}$$

8.3. Ranking model

Figure 8.1 demonstrates how “learning to rank” works. We train the supervised learning to rank with training dataset. Both the training and test set contain the same features sets; the test set does not include ranking judgements, which will be predicted by the model learned

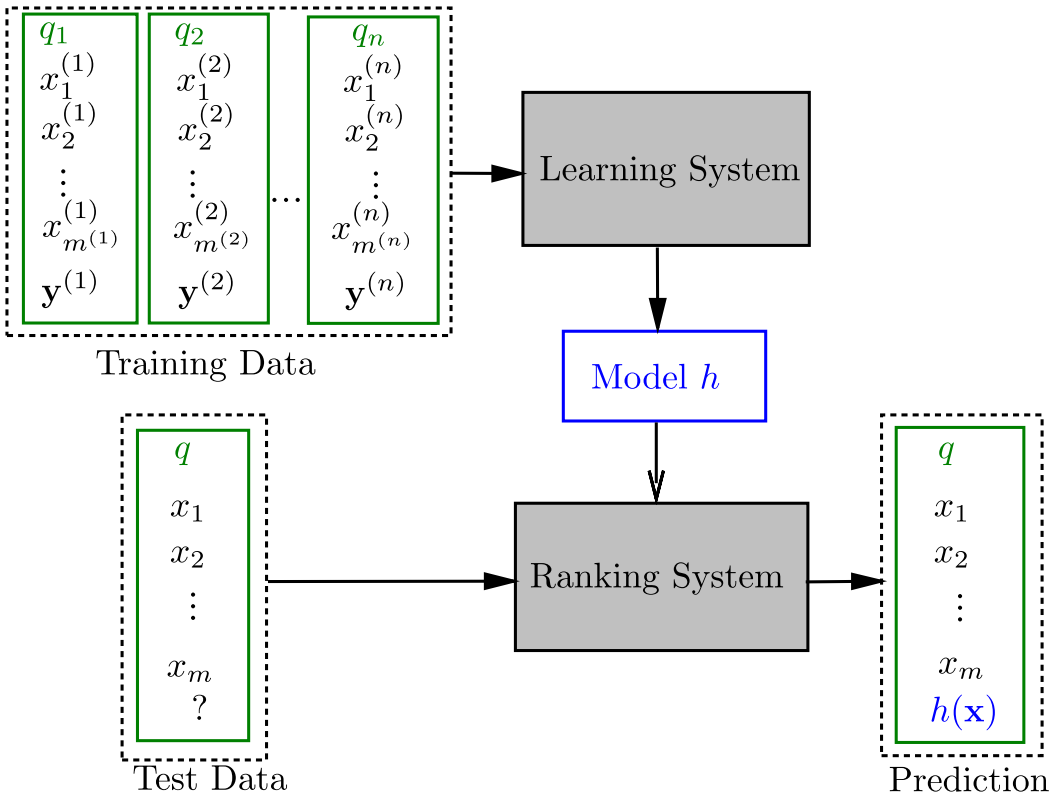

Figure 8.1.. Learning-to-rank framework (taken from Liu [59]).

from training set. For the passage ranking task, a typical training set comprises of n training questions q_i $(i = 1, \ldots, n)$ and their associated passages; they are represented by feature vectors $\mathbf{x}^{(i)} = \{x_j^{(i)}\}_{j=1}^{m^{(i)}}$ (where $m^{(i)}$ denotes the number of passages relevant to question q_i). With those aforementioned features, we employ a specific learning algorithm to learn the ranking model. During the test phase, the learned model will sort the passages regarding their relevance to the question, and return a sorted list of passages, which will be used for answer extraction.

The major approaches for learning to rank can be categorized as:

Pointwise approaches treat the ranking problem as a classification or regression problem on individual instances in binary labelled data and rank each instance according to predicted scores.

Pairwise approaches learn to optimize the pairwise preference between passage pairs, and therefore formalize the ranking problem as the classification of binary preference between passage pairs.

Listwise approaches consider the overall ranking of an entire group of passages relevant to a question and define the loss function that optimizes the overall ranking of passages for one question.

In this section, we briefly describe three representative machine learning algorithm under the framework of Support Vector Machines (SVMs) [96], which has been proven to be one of the best machine learning models in many applications.

8.3.1. Support Vector Regression

Support Vector Regression (SVR) [87] is the application of SVM in the case of regression problems. It maintains all the main features that characterise the maximal margin algorithm, which gives it a good generalization ability. The kernel functions are also introduced to SVR to handle complex non-linear problems.

Given the training data $\mathbf{x} = \{\phi(q, p_j), y_j\}_{j=1}^{m}$ for the question q, ϕ denotes the feature functions which we defined in Section 8.2 for each pair of q and p_i. $\mathbf{y} = \{y_j\}_{j=1}^{m}$ are binary relevance judgements associated with question q. $y_j = 1$ denotes the supporting passages as positive examples, otherwise $y_j = 0$ as negative examples.

We can formalize this regression problem as a convex optimization problem:

$$\begin{aligned} \min_{w,\xi \geq 0} \ & \frac{1}{2}||w||^2 + C\sum_i(\xi_i + \xi_i^*) \\ s.t. \ & y_i - \omega \cdot \phi(q, p_i) - b \leq \varepsilon + \xi_i \\ & \omega^T \phi(q, p_i) + b - y_i \leq \varepsilon + \xi_i^* \end{aligned} \tag{8.23}$$

where w, b are the weights parametrizing the hyperplanes that separate the positive

from negative training data. Slack variables ξ_i, ξ_i^* are introduced to cope with otherwise infeasible constraints of the optimization problem. The parameter C is a regularization term, which controls the trade-off between minimizing $\|w\|^2$ and penalty function $\sum \xi_i$. ε is a free constraint for the error rate of each instance.

SVR is a pointwise approach with a ranking function designed to estimate the accurate relevance degree of individual objects. Such a formulation is limited, as we would not be able to model the redundancy and preference among passages. In the following section we will introduce pairwise approaches to modelling the relative orders of a pair of objects.

8.3.2. Ranking SVM

Ranking SVM [45] is a pairwise SVM algorithm learning from pairwise reference of objects rather than individual objects. For the question q, $\mathcal{C}^q$ and $\mathcal{C}^{\bar{q}}$ denote the set of supporting and non-supporting passages of $\mathcal{C}$. A linear scoring function $f(x) = \omega^T \phi(q, p_i)$ is used to rank the passages.

Given two passages (p_u, p_v), $p_u \in \mathcal{C}^q$ and $p_v \in \mathcal{C}^{\bar{q}}$, p_u shall be ranked ahead of p_v, therefore $f(\phi(q, p_u)) > f(\phi(q, p_v))$.

$$\begin{aligned} p_u \succ p_v \quad &\Leftrightarrow \quad f(\phi(q, p_u)) \succ f(\phi(q, p_v)) \\ &\Leftrightarrow \quad \omega^T(\phi(q, p_u) - \phi(q, p_v)) > 0 \end{aligned} \tag{8.24}$$

Let $y_{u,v}$ be the pairwise difference between features $\phi(q, p_u) - \phi(q, p_v)$. One can convert the ranking problem into classification problem by assigning new label $y_{u,v}$.

$$y_{u,v} = \begin{cases} +1 & \text{if } p_u \succ p_v \\ -1 & \text{if } p_u \prec p_v \end{cases} \tag{8.25}$$

$y_{u,v} = +1$ if a relevant passage p_u ranked ahead of an irrelevant passage p_v, and $y_{u,v} = -1$

if p_j ahead of p_i. The ranking SVM model is formulated as follows:

$$\begin{aligned} \min_{w,\xi \geq 0} \quad & \frac{1}{2}||w||^2 + C \sum \xi^q_{u,v} \\ s.t. \quad & y^q_{u,v} \cdot \omega^T(\phi(q, p_u) - \omega \cdot \phi(q, p_v)) \geq 1 - \xi^q_{u,v} \\ & \xi^q_{u,v} \geq 0 \end{aligned} \tag{8.26}$$

The object function in Ranking SVM takes the same formulation as in SVR. The difference between them is the constraints, which depends on the passage pairs and corresponding preference labels.

8.3.3. SVM^{map}

Xia et al. [114] point out that listwise learning-to-rank approaches usually outperform pointwise and pairwise approaches. The listwise approaches attempt to learn the ordering information from a entire group of passages associated with a question, and directly optimize a continuous or differentiable bound of the IR evaluation metrics. We employ the listwise SVM^{map} [119], which optimizes the IR evaluation measure — mean average precision (MAP), defined as

$$\mathrm{MAP}(r, \hat{r}) = \frac{1}{rel} \sum_{j:r_j=1} Prec@j, \tag{8.27}$$

where rel is the number of supporting passages, and $Prec@j$ is the proportion of supporting passages in the top j passages of the predicted ranking $\hat{\mathbf{y}}$. MAP is the average over all test questions.

We first give a definition of the listwise ranking task. The input space includes a set of passages for a question q, the output is a ranked list of passages, and the ranking function $h\colon \mathcal{X} \to \mathcal{Y}$ maps the input space $\mathcal{X}$ (unordered list of passages) to output space $\mathcal{Y}$ (all rankings of passages).

Firstly, SVM^{map} directly optimizes mean average precision (MAP), a widely used evaluation of overall performance of IR systems. In doing so, we define the loss function $\Delta\colon \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}$. $\Delta(\mathbf{y}, \hat{\mathbf{y}})$ qualifies the penalty for predicting the ranking $\hat{\mathbf{y}}$ given the gold-standard ranking $\mathbf{y}$ for one question. Let $r = rank(\mathbf{y})$ and $\hat{r} = rank(\hat{\mathbf{y}})$, then

$$\Delta_{map}(\mathbf{y}, \hat{\mathbf{y}}) = 1 - \mathrm{MAP}(r, \hat{r}). \tag{8.28}$$

Secondly, the listwise feature function $\Psi(q, \mathbf{y})$ for the question q and passage rankings $\mathbf{y}$ is the aggregate over the feature vector differences of all supporting/non-supporting passages pairs.

$$\Psi = \frac{1}{|C^q| \cdot |C^{\bar{q}}|} \sum_{p_u \in C^q} \sum_{p_v \in C^{\bar{q}}} [y_{u,v}(\phi(q, p_u) - \phi(q, p_v))] \tag{8.29}$$

The notation is the same as in Ranking SVM.

In the binary relevance scenario, rankings are represented as a matrix of pairwise orderings $\mathcal{Y} \subset \{-1, 1\}^{|\mathcal{C}| \times |\mathcal{C}|}$, The optimization problem of SVM^{map} is formulated as

$$\begin{aligned} \min_{w, \xi \geq 0} \ & \frac{1}{2}||w||^2 + \frac{C}{n} \sum_i \xi_i \\ s.t. \ & \forall i, \forall \mathbf{y} \in \mathcal{Y} \backslash \mathbf{y}_i : \\ & \omega^T \Psi(q_i, \mathbf{y}_i) \geq \omega^T \Psi(q_i, \mathbf{y}) + \Delta(\mathbf{y}_i, \mathbf{y}) - \xi_i \end{aligned} \tag{8.30}$$

8.4. Experimental Results

The main goal of our experiments is to test the feasibility of leveraging learning to rank methods with crowdsourcing annotations. We design experiments to test the effectiveness of those representative learning to rank models on supporting passages for list questions. The first experiment evaluates the ranking accuracy using all features we described. In the second experiments, we study the contributions of each feature by incrementally adding

them to training and testing learning to rank approaches. Then we test how those models perform on retrieved documents with the presence of a large proportion of irrelevant and non-supporting documents.

8.4.1. Datasets

We created the passage ranking dataset for list questions by crowdsourcing annotation collection and learning described in Chapter 6 and 7. We select 51 list questions from TREC2005 as training data and 33 list questions from TREC2004 as test data. Before feature extraction, we carry out the following preprocessing steps of question and passage:

- Stop words are removed, and Porter stemming [6] is used before calculating the **BOW** and **proximity** features.
- Stanford NER tool [7] recognizes NEs for calculating **NE proximity** features.
- WordNet module in NLTK [8] measures WordNet-based **semantic** features.
- The xgoogle [9] tool is employed to extract the search statistics and snippets from search results for **web popularity** and **kernel** features.
- Feature scaling with L^2 norm is applied to all feature vectors.

We use the SVMlight [10] implementation of ranking SVMs and SVR and the standard implementation of SVM^{map} [11] [119].

[6] http://tartarus.org/~martin/PorterStemmer/
[7] http://nlp.stanford.edu/ner/
[8] http://www.nltk.org/
[9] www.catonmat.net/blog/python-library-for-google-search/
[10] http://svmlight.joachims.org/
[11] http://projects.yisongyue.com/svmmap/

8.4.2. Evaluation Measures

As QA systems usually select top N passages to extract answers, we use the prediction accuracy, i.e., the proportion of supporting passages at top 5, 20 and 100 passages as a evaluation score. The precision at a given cutoff *k* ($P@k$) is defined as

$$P@k(q) = \frac{\#\{\text{supporting passages in the top } k \text{ positions}\}}{k} \tag{8.31}$$

$P@k$ does not average well across questions since questions have different numbers of passages from relevant documents.

We also interest in the effectiveness of our methods on the ranking an entire group of passages. We thus use the mean average precision (MAP) defined in Equation 8.28.

8.4.3. Baselines

We run the retrieval algorithms implemented in Indri tools [12] and use the best results as the baseline. We also compared with a cluster-based language model with co-occurrence statistics from the Google n-gram corpus, the best model in the existing state of the art [71].

The algorithms from Indri include:

- **TFIDF Model**. Both passages and questions are represented as vectors (denoted by $\boldsymbol{p}$ and $\boldsymbol{q}$), each element of which represents the TFIDF weight of a token. The similarity between $\boldsymbol{p}$ and $\boldsymbol{q}$ defined as

$$\text{sim}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{n} tf(t_i, p) \cdot tf(t_i, q) \cdot idf(t_i)^2, \tag{8.32}$$

  is used as the ranking score.

- **Vector Space Model**. The ranking score is calculated as the cosine similarity

[12] http://www.lemurproject.org/indri.php

between $\boldsymbol{p}$ and $\boldsymbol{q}$,

$$\text{sim}(\boldsymbol{p}, \boldsymbol{q}) = \frac{\sum_{i=1}^{n} tfidf(t_i, p) \cdot tfidf(t_i, q)}{\sqrt{\sum_{i=1}^{n} tfidf(t_i, p)^2} \cdot \sqrt{\sum_{i=1}^{n} tfidf(t_i, q)^2}} \tag{8.33}$$

- **Okapi B25 Model**. The probabilistic ranking model rank documents by the log-odds of their relevance.

$$\text{BM25}(\boldsymbol{p}, \boldsymbol{q}) = \sum_{i=1}^{n} \frac{idf(t_i) \cdot tf(t_i, p) \cdot (k_1 + 1)}{tf(t_i, p) + k_1 \cdot (1 - b + b \cdot \frac{\text{LEN}(p)}{\text{avgdl}})}, \tag{8.34}$$

  where **LEN(p)** is the length of a document in words. **avgdl** is the average document length in the text collection from which documents are drawn. k_1 and b are free parameters. Other variables have the same meaning in the previous function. Passages are ranked regarding the **B25** score.

- **Language Model (LM)**. The basic idea is to estimate a language model for each passage and rank passages by the likelihood of query tokens give by the language model. Given the assumption of the independence among terms,

$$p(\boldsymbol{q}|\boldsymbol{p}) = \prod_{i=1}^{n} p(t_i|\boldsymbol{p}) \tag{8.35}$$

  To adjust the maximum likelihood estimation of unigram probability $p(t_i|\boldsymbol{p})$, smoothing techniques [121] are introduced to deal with data sparseness.

- **Kullback-Leibler (KL) Divergence Algorithm**. Let θ_q be the language model for the query q and θ_p be the language model for passage p. The passages are ranked by $-D(\theta_q|\theta_p)$, where function D denotes the KL divergence,

$$D(\theta_q|\theta_p) = \sum_{i=1}^{n} p(t_i|\theta_q) \log \frac{p(t_i|\theta_q)}{p(t_i|\theta_p)} \tag{8.36}$$

  The Indri tool provides maximum likelihood estimation with various smoothing methods for calculating the language model $P(t_i|\theta)$.

Momtazi and Klakow [71] introduced term clustering into LM and formalized the class-based LM to overcome the problem of data sparsity and exact matching in text retrieval for QA. Given question term clusters C_q, the $p(\boldsymbol{q}|\boldsymbol{d})$ is further derived as following,

$$p(\boldsymbol{q}|\boldsymbol{p}) = \prod_{i=1}^{M} P(t_i|C_{t_i}, \boldsymbol{p}) p(C_{t_i}|\boldsymbol{p}), \tag{8.37}$$

where M is the number of term clusters. $p(C_{t_i}|\boldsymbol{p})$ is similar to $p(t_i|\boldsymbol{p})$ in Equation 8.35, however it is calculated on clusters instead of terms. To obtain C_q, they used a word clustering algorithm based on word co-occurrence statistics for large-scale corpora. Words within a cluster can be viewed as sharing certain semantic properties.

8.4.4. Results

In this section, we experimentally compare the effectiveness of SVR, ranking SVM and SVM^{map} and baselines on the gold standard dataset. Table 8.1 summarizes statistics for training and test datasets, which indicates that the number of documents, passages, and relevant passages for each question is quite varied.

	training	**testing**
Q/Doc Pairs	1160/70/1	780/71/1
Q/P Pairs	13K/1072/5	9853/861/18
supporting P	1570/96/1	1295/153/1

Table 8.1.. Overview of the datasets used in the supporting documents evaluation. Q: question, Doc: document, and P: passages. Each element contain 3 values: total number of relevant passages / maximum number of relevant passages for a question / minimum number of relevant passages for a question.

Table 8.2 compares the performance of different models on ranking passages for list questions. Comparing the learning to rank SVM methods with the best retrieval model and Momtazi et al.'s class-based LM, we find that all SVM-based ranking models significantly outperform the baselines, which is not surprising, as learning to rank algorithms consider various feature representations.

Regarding the evaluation of top rankings, we see that SVM^{map} is better than SVR (4.9% improvement of P@100), and that SVR model performs statistically significantly better than ranking SVMs (20.11% improvement of P@100). Considering the MAP metric, SVR model is better that SVM^{map} and ranking SVMs model. Thus for passage ranking, SVM^{map} is robust and highly effective in ranking top results as it uses listwise features across different passages and distinguishes between supporting passages and non-supporting passages. Moreover Momtazi's method needs a longer time to extract the term co-occurrence from the large-scale corpus, which takes several days on our workstation, whereas on average our method takes around two hours for feature extraction and ten minutes on training a model.

model	MAP	P@5	P@20	P@100
SVM^{map}	0.6250	**0.6606**	**0.4642**	**0.2148**
Momtazi et al.	0.5332	0.5321	0.3971	0.1766
Ranking SVM	0.6185	0.6242	0.4424	0.2045
SVR	0.6273	0.6303	0.4591	0.2018
basline	0.3687	0.3818	0.2970	0.1612

Table 8.2.. Performance comparison of passage selection models

Feature Set	Feature Count	SVM^{map}		ranking SVMs		SVR	
		MAP	P@20	MAP	P@20	MAP	P@20
BOW	2	0.3379	0.2833	**0.3866**	**0.3167**	0.3517	0.2939
+Context BOW	2	0.3391	0.2818	**0.5180**	**0.3773**	0.3758	0.3030
+Proximity	1	0.3989	0.3561	0.4302	0.3712	**0.4728**	**0.4076**
+NE Proximity	4	0.6205	**0.4652**	0.6195	0.4515	**0.6294**	0.4561
+Web	1	0.6221	**0.4638**	0.6226	0.4530	**0.6274**	0.4561
+Semantic	2	0.6250	**0.4642**	0.6185	0.4424	**0.6273**	0.4591

Table 8.3.. Performance of ranking models using different individual and incremental feature groups.

Table 8.3 reports the contributions of each feature set. During the training and test phase, we initialize with simple *BOW* features, and successively add feature sets that provide the highest MAP improvement on the test data.

The adding of *contextual BOW* feature sets shows quite a noticeable improvement over the BOW features for ranking SVM model (34.20% improvement of MAP), however the NE Proximity features decrease the performance of ranking SVM, while they bring greatest improvement of the SVM^{map} (55.55% improvement of MAP) and SVR model (33.12% improvement of MAP).

The web-based features show nearly no help in improving overall performance (CCP and PMI do not increase overall performance). Semantic features slightly improve of SVM^{map} and decrease the performance of ranking SVM and SVR, which is not effective w.r.t. the complicated computation. The results indicate that it is important to select the ad-hoc feature sets for different learning to rank models.

8.4.5. Performances on Retrieved Documents

In the real QA scenario, passage retrieval model takes relevant documents produced by document/passage retrieval as input. There is no guarantee that a document definitely contains an answer to a question. To evaluate how our passage retrieval models perform as a part of a QA system, we merge the top 50 retrieved documents provided by TREC [13] with the gold standard documents as the retrieved document set, on which we test the influence of irrelevant and non-supporting documents on the performance of passage ranking. Table 8.4 summarizes statistics for training and test datasets, which is much sparser than those in Table 8.1.

	training	**testing**
Q/Doc Pairs	3081/119/11	1961/97/50
Q/P Pairs	44K/2010/179	30K/1410/481

Table 8.4.. Overview of the data used in the retrieved documents evaluation.

We choose the most effective model—SVM^{map} and compare it with the baseline IR model to show the robustness of learning to rank models. Comparing results in Table 8.5 and

[13] http://trec.nist.gov/data/qa/

model	MAP	P@5	P@20	P@100
SVM^{map}	**0.4451**	**0.4788**	**0.3485**	**0.1879**
baseline	0.1423	0.1636	0.1439	0.0855

Table 8.5.. Performance comparison on retrieved documents data.

Table 8.2, both SVM^{map} and baseline (TFIDF model) show lower performance on the large and noisy retrieved dataset. The baseline retrieval model is more sensitive to irrelevant passages than SVM^{map}, as those lexical matching models bias towards non-supporting passages containing more question tokens, which are certainly relevant passages instead of supporting passages.

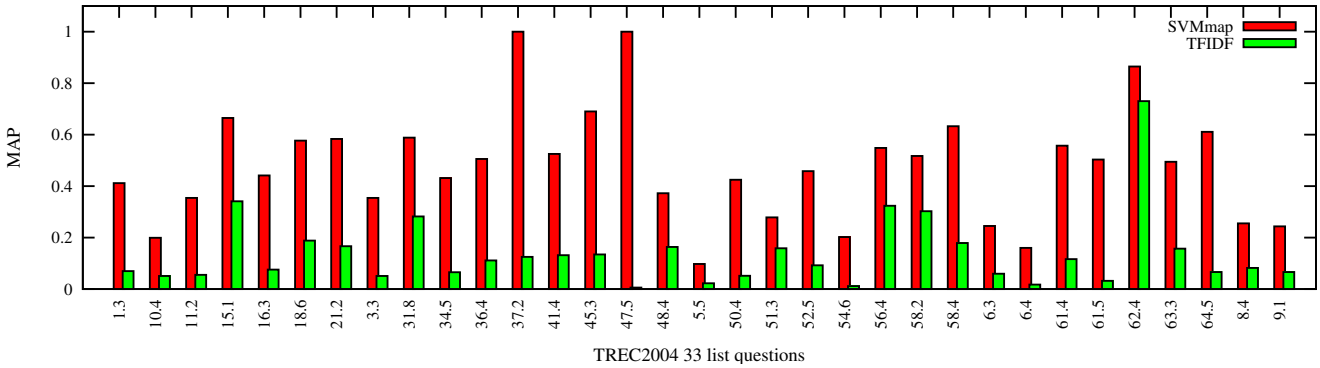

Figure 8.2.. Comparison of SVM^{map} and baseline on the TREC 2004 set of 33 questions.

A detailed comparison on each question in Figure 8.2 shows that SVM^{map} greatly outperforms the baseline for all questions.

8.5. Conclusions

In this chapter, we have discussed a novel learning method to rank supporting passages for TREC list questions. By applying effective SVM-based learning-to-rank models, we provided a framework to leverage features that measure the question-passage relevance from various linguistic aspects. Our approaches can improve the passage ranking results significantly over retrieval models and other state-of-the-art method. The experimental

results also validated the contributions of features and indicated the importance of feature selection for learning to rank techniques.

9. Conclusions

9.1. Summary

Knowledge acquisition has been one of the most important activities in human history. Recent decades with the explosive growth of online data and information, many effective and intelligent knowledge acquisition systems have been developed for assisting people in searching information and learning knowledge.

Virtually all supervised knowledge acquisition methods heavily rely on annotated data. Therefore, annotation acquisition bottleneck is undoubtedly one of the most important issues in building knowledge acquisition systems. We already discussed in this thesis two key techniques on leveraging online collaborative work for alleviating this problem: mining from online collaborative knowledge repositories and creating ad-hoc annotations with crowdsourcing.

The first strategy investigates how to extract or derive structured content from Wikipedia. Wikipedia is composed of consistent presentation and connection of information about entities. We build cross-lingual entity linking (CLEL) systems for enriching entities with those knowledge base (KB) in Chapter 3 and 4. The matching of entities and KB is not trivial due to the language barrier, entity synonymy and polysemy. To solve the first two problems, we further extract cross-lingual taxonomy and expand each KB nodes with linking and redirecting information from Wikipedia. We adopt two methods for resolving ambiguous entities. The IR-based method selects the KB node with the most relevant Wikipedia articles as target linkings, whereas the generative CLEL model takes advantage

of global entity popularity from the Wikipedia and local contextual relevance from background documents. The generative model performs much better than our original CLEL system, and can rank at the second position comparing with participating systems in 2011.

The second strategy is more flexible and task-oriented. The Wikipedia KB is of high quality and quantity, however it is not trivial work to adjust or modify Wikipedia KB for specific tasks. The crowdsourcing strategy can get things done by outsourcing the task to a large crowd of online workers. The fast accumulation of crowd wisdom contributes to large-scale processing of data. We use crowdsourcing to create annotations for question answering (QA) systems.

The object of QA is to pinpoint answers to a question from relevant documents. The passage ranking model is a crucial step from documents to answers. It can largely reduce the search space for answer extraction, from a complete document to precise and compact answer-bearing passages. To create supporting passages for QA, we propose the framework of learning with crowdsourced annotation depicted in Figure 9.1.

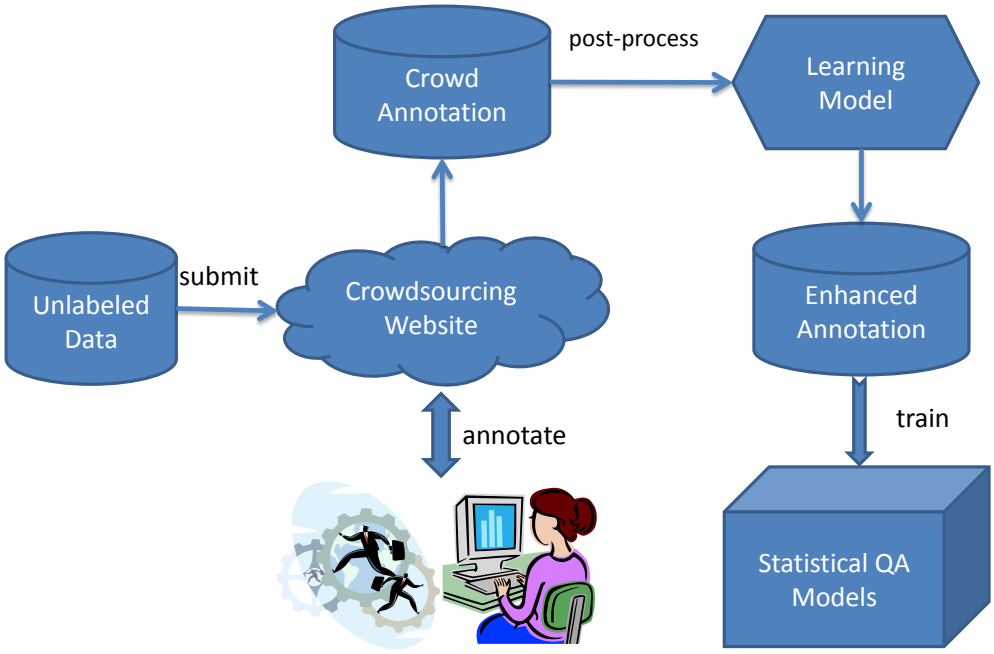

Figure 9.1.. Learning with Crowd Annotation.

1. We design online experiments and submit unlabelled data onto Amazon's Mechanical Turk (AMT) as micro-tasks in Chapter 6, so that online workers can take part in annotating work.
2. Crowd annotations are noisy. We investigate true label learning models in Chapter 7 to enhance the quality of crowd annotations.
3. Those enhanced annotations are used for training QA tasks, such as passage ranking in Chapter 8.

Our online experiments suggest proper settings of the approval rate and the number of workers can lead to better crowd annotations. For the true label learning, supervised models performs the best however unsupervised models can still be practical for enhancing crowd annotations when gold standard annotations are not available.

The learning to rank models trained on enhanced crowd annotations, were shown to be effective for passage ranking for list questions. Our learning to rank methods based on a variety of features outperformed a start-of-the-art model by increasing MAP of 17.2% on list questions.

9.2. Future Work

Our methods on CLEL are language-independent. It is applicable to apply for experimenting with multiple languages, such as German-English, French-English and English-Japan CLEL. The English Wikipedia can also use as a bride language to transform linking information between pairs of languages. Our methods rely heavily on the cross-lingual taxonomy and linkages extracted from bilingual versions of Wikipedia, however those resources are very limited for many low-resource languages, which involve a small amount of Wikipedia pages. Therefore it is a challenging task to link entities in low-resource languages with English Wikipedia articles. On possible solution is to extend the interlingual representation approach introduced in Section 2.3.4. In addition to local entities in the

background documents, we can retrieve relevant entities in a low-resource language version of Wikipedia and find their global corresponding entities in English Wikipedia. Both local and global entities can be jointly used to infer the linked entities with graphical models.

Those two strategies can be easily transferred to building cross-lingual slot filling (CLSL). SL is to find the values of specified attributes ("slot") of the entity from a large collection of source documents, such as the birthday and children of a person or the website and employees of an organization. We can use the similar strategies of CLEL for CLSL. CLSL needs knowledge evidence from Wikipedia and contextual evidence for selecting slot candidates, determine the relations between an entity and its slot candidates, and choose the most confident slot value. One critical problem for SL is the lack of annotations for building relation extraction methods. As a case study, we can easily transfer the annotation of a pair of entity and slot into crowdsourcing tasks.

1. **Finding slot entities and sentences containing those entities.** We convert the requirement of determining a slot value for an entity into an information search problem. For example, given the entity "Microsoft" and slot value "top employee", we design a crowdsourcing task with the description: "*Please refer to Wikipedia pages, find out who is the CEO of Microsoft and a sentence which supports your answer. Please input the answer and sentence separately in text boxes*". We can ask many other questions, such as Wikipedia pages for each entities and the translation of entities.

2. **Judging those relation sentences.** Those relation sentences created in Task 1 may contain adverse or ambiguous annotations, therefore online workers are required to judge the correctness of those sentences in the second task: "*Please judge whether the sentence supports the argument that **Bill Gates** is **CEO** of **Microsoft***.".

Mechanical Turk supports multiple interactions with online workers. Its versatile features make it possible to create annotations in different levels and forms.

Another interesting direction for future work is to combine these two approaches. Many researches work on extracting or deriving structured knowledge from Wikipedia to support specific tasks. The process of data extraction and checking is tedious and time-consuming. Crowdsourcing annotation and judgement can be a good choice for conducting those pre-processing work.

Last but not least, More attention shall be paid in design of crowdsourcing experiments, effective interactions with online users, and methods for enhancing crowd annotations. Boosting the quality of crowdsourcing is beneficial for improving the overall reputation of crowdsourcing markets and accelerating the procedure of original research work in NLP and IR.

Bibliography

[1] Enek Agirre, Angel X. Chang, Daniel S. Jurafsky, Christopher D. Manning, Valentin I. Spitkovsky, and Eric Yeh. Stanford-UBC at TAC-KBP. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[2] Omar Alonso, Michael Gertz, and Ricardo Baeza-Yates. Clustering and Exploring Search Results Using Timeline Constructions. In *Proceedings of CIKM*, Hong Kong, China, 2009.

[3] Omar Alonso and Stefano Mizzaro. Can We Get Rid of TREC Assessors? Using Mechanical Turk for Relevance Assessment. In *SIGIR '09: Workshop on The Future of IR Evaluation*, 2009.

[4] Ivo Anastácio, Bruno Martins, and Pável Calado. Supervised Learning for Linking Named Entities to Knowledge Base Entries. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[5] Ion Androutsopoulos. Natural Language Interfaces to Databases - An Introduction. *Journal of Natural Language Engineering*, 1:29–81, 1995.

[6] Javier Artiles, Julio Gonzalo, and Satoshi Sekine. The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. In *Proceedings of SemEval-2007*, pages 64–69. Association for Computational Linguistics, June 2007.

[7] Somnath Banerjee, Soumen Chakrabarti, and Ganesh Ramakrishnan. Learning to Rank for Quantity Consensus Queries. In *Proceedings of SIGIR*, pages 243–250,

2009.

[8] Jeff Barr and Luis-Felipe Cabrera. AI Gets a Brain. *ACM Queue*, 4(4):24–29, 2006.

[9] Dan Bikel, Vittorio Castelli, Radu Florian, and Ding-Jung Han. Entity Linking and Slot Filling through Statistical Processing and Inference Rules. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[10] Chris Callison-burch. Fast, Cheap, and Creative: Evaluating Translation Quality Using Amazonś Mechanical Turk. In *Proceedings of EMNLP*, Singapore, 2009.

[11] Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. Learning to rank: from pairwise approach to listwise approach. In *ICML '07*, pages 129–136. ACM, 2007.

[12] Taylor Cassidy, Zheng Chen, Javier Artiles, Heng Ji, Hongbo Deng, Lev-Arie Ratinov, Jing Zheng, Jiawei Han, and Dan Roth. CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[13] David Chen and William Dolan. Collecting Highly Parallel Data for Paraphrase Evaluation. In *Proceedings of the 49th ACL: HLT*, pages 190–200, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[14] Zheng Chen and Heng Ji. Collaborative ranking: a Case Study on Entity Linking. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 771–781, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[15] Zheng Chen, Suzanne Tamang, Adam Lee, Xiang Li, Wen-Pin Lin, Matthew Snover, Javier Artiles, Marissa Passantino, and Heng Ji. CUNY-UIUC-SRI TAC-KBP2011 Entity Linking System Description. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[16] David Chiang. Hierarchical Phrase-based Translation. *Computational Linguistics*, 33, 2007.

[17] Grzegorz Chrupała, Saeedeh Momtazi, Michael Wiegand, Stefan Kazalski, Fang Xu, Benjamin Roth, Alexandra Balahur, and Dietrich Klakow. Saarland University Spoken Language Systems at the Slot Filling Task of TAC KBP 2010. In *Proceedings of TAC 2010*, 2010.

[18] Rudi L. Cilibrasi and Paul M. B. Vitanyi. The Google Similarity Distance. *IEEE Trans. on Knowl. and Data Eng.*, 19(3):370–383, 2007.

[19] Andras Csomai and Rada Mihalcea. Wikify! Linking Documents to Encyclopedic Knowledge. In *Proceedings of CIKM07*, volume 23, pages 34–41, 2007.

[20] Silviu Cucerzan. Large-Scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of EMNLP-CoNLL*, pages 708–716, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[21] Silviu Cucerzan. TAC Entity Linking by Performing Full-document Entity Extraction and Disambiguation. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[22] Dan Shen and Jochen L. Leidner and Andreas Merkel and Dietrich Klakow. The Alyssa System at TREC 2006: A Statistically-Inspired Question Answering System. In *Fifteenth Text REtrieval Conference (TREC 2006)*, 2006.

[23] Hoa Trang Dang, Diane Kelly, and Jimmy J. Lin. Overview of the TREC 2007 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2007*, 2007.

[24] Hoa Trang Dang, Jimmy J. Lin, and Diane Kelly. Overview of the TREC 2006 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2006*, 2006.

[25] Hoa Trang Dang and Karolina Owczarzak. Overview of the TAC 2008 Opinion Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2008*, 2008.

[26] Jia Deng, Wei Dong, Richard Socher, Li jia Li, Kai Li, and Li Fei-fei. Imagenet: A

Large-Scale Hierarchical Image Database. In *Proceedings of CVPR*, 2009.

[27] Angela Fahrni and Michael Strube. HITS' Cross-lingual Entity Linking System at TAC2011: One Model for All Languages. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[28] David A. Ferrucci, Eric W. Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John M. Prager, Nico Schlaefer, and Christopher A. Welty. Building Watson: An Overview of the DeepQA Project. *AI Magazine*, 31(3):59–79, 2010.

[29] Jenny Rose Finkel, Trond Grenager, and Christopher Manning. Incorporating Non-Local Information into Information Extraction Systems by Gibbs Sampling. In *Proceedings of the 43rd ACL*, pages 363–370. Association for Computational Linguistics, 2005.

[30] Jun'ichi Fukumoto, Tsuneaki Kato, and Fumito Masui. Question Answering Challenge (QAC-1) An Evaluation of Question Answering Task at NTCIR Workshop 3. In *NTCIR Workshop 3*, 2003.

[31] Evgeniy Gabrilovich and S. Markovitch. Wikipedia-based Semantic Interpretation for Natural Language Processing. *Journal of Artificial Intelligence Research*, 34(1):443–498, 2009.

[32] Bert F. Green, Jr., Alice K. Wolf, Carol Chomsky, and Kenneth Laughery. Baseball: an automatic question-answerer. In *Papers presented at the May 9-11, 1961, western joint IRE-AIEE-ACM computer conference*, IRE-AIEE-ACM '61 (Western), pages 219–224, New York, NY, USA, 1961. ACM.

[33] Xianpei Han and Le Sun. A Generative Entity-mention Model for Linking Entities with Knowledge Base. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 945–954, Stroudsburg, PA, USA, 2011. Association for Computational Linguistics.

[34] Xianpei Han and Jun Zhao. NLPR_KBP in TAC 2009 KBP Track: A Two–Stage Method to Entity Linking. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[35] David Heckerman, David Maxwell Chickering, Christopher Meek, Robert Rounthwaite, and Carl Kadie. Dependency Networks for Inference, Collaborative Filtering, and Data Visualization. *J. Mach. Learn. Res.*, 1:49–75, September 2001.

[36] Paul Heymann and Hector Garcia-Molina. Turkalytics: Analytics for Human Computation. In *Proceedings of the 20th international conference on World wide web*, WWW '11, pages 477–486, New York, NY, USA, 2011. ACM.

[37] M. Honnibal and R. Dale. DAMSEL: The DSTO/Macquarie System for Entity-Linking. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[38] Jeff Howe. Crowdsourcing: A definition. http://crowdsourcing.typepad.com/cs/2006/06/crowdsourcing_a.html, 2006. [Online; accessed 29-June-2012].

[39] Zhiheng Huang, Marcus Thint, Zengchang Qin, Intelligent Systems, and Research Centre. Question Classification using Head Words and their Hypernyms. In *Proceedings of EMNLP*, 2008.

[40] Barbara J. Grosz, Spärck-Jones Karen, and Bonnie L. Webber. *Readings in Natural Language Processing*. Morgan Kaufmann, 1986.

[41] Heng Ji, Ralph Grishman, and Hoa Trang Dang. Overview of the TAC 2011 Knowledge Base Population Track. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[42] Heng Ji, Ralph Grishman, Hoa Trang Dang, Kira Griffitt, and Joe Ellis. Overview of the TAC 2010 Knowledge Base Population Track. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[43] Xu Jian, Hector Liu, Qin Lu, Patty Liu, and Chen Chen Wang. PolyUCOMP in TAC

2011 Entity Linking and Slot-Filling. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[44] Jay J. Jiang and David W. Conrath. Semantic Similarity based on Corpus Statistics and Lexical Taxonomy. In *Proc. of the Int'l. Conf. on Research in Computational Linguistics*, pages 19–33, 1997.

[45] Thorsten Joachims. Optimizing Search Engines Using Clickthrough Data. In *Proceedings of the eighth ACM SIGKDD*, pages 133–142, New York, NY, USA, 2002. ACM.

[46] Michael Kaisser, Marti A. Hearst, and John B. Lowe. Improving Search Results Quality by Customizing Summary Lengths. In *Proceedings of ACL-08: HLT*, pages 701–709, Columbus, Ohio, June 2008. Association for Computational Linguistics.

[47] Michael Kaisser and John B. Lowe. Creating a Research Collection of Question Answer Sentence Pairs with Amazon's Mechanical Turk. In *Proceedings of International Conference on Language Resources and Evaluation, LREC*, Marrakech, Morocco, 2008.

[48] Philipp Koehn. *Statistical Machine Translation*. Cambridge University Press, 2009.

[49] Julian Kupiec. MURAX: a Robust Linguistic Approach for Question Answering Using an On-line Encyclopedia. In *SIGIR '93*, pages 181–190. ACM, 1993.

[50] John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *ICML '01*, pages 282–289. Morgan Kaufmann Publishers Inc., 2001.

[51] Victor Lavrenko and W. Bruce Croft. Relevance Based Language Models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '01, pages 120–127, New York, NY, USA, 2001. ACM.

[52] Florian Laws and Christian Scheible. Active learning with Amazon Mechanical Turk.

In *Proceedings of EMNLP*, pages 1546–1556, 2011.

[53] Claudia Leacock and Martin Chodorow. *Combining Local Context and WordNet Similarity for Word Sense Identification*, chapter 11, pages 265–283. The MIT Press, 1998.

[54] John Lehmann, Sean Monahan, Luke Nezda, Arnold Jung, and Ying Shi. LCC Approaches to Knowledge Base Population at TAC 2010. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[55] Fangtao Li, Zhicheng Zheng, Fan Bu, Yang Tang, Xiaoyan Zhu, and Minlie Huang. THU QUANTA at TAC 2009 KBP and RTE Track. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[56] Qi Li, Sam Anzaroot, Wen-Pin Lin, Xiang Li, and Heng Ji. Joint inference for cross-document information extraction. In *CIKM '11*, pages 2225–2228. ACM, 2011.

[57] Dekang Lin. An Information-Theoretic Definition of Similarity. In *Proceedings of the 15th ICML*, pages 296–304, 1998.

[58] Greg Little, Lydia B. Chilton, Max Goldman, and Robert C. Miller. TurKit: Tools for Iterative Yasks on Mechanical Turk. In *HCOMP '09: Proceedings of the ACM SIGKDD Workshop on Human Computation*, pages 29–30, New York, NY, USA, 2009. ACM.

[59] Tie-Yan Liu. Learning to Rank for Information Retrieval. *Found. Trends Inf. Retr.*, 3(3):225–331, March 2009.

[60] Yuanhua Lv and ChengXiang Zhai. Positional Language Models for Information Retrieval. In *Proceedings of SIGIR*, pages 299–306, New York, New York, USA, 2009. ACM Press.

[61] Bernardo Magnini, Matteo Negri, Roberto Prevete, and Hristo Tanev. Is It the Right Answer? Exploiting Web Redundancy for Answer Validation. In *Proceedings of ACL*, pages 425–432, 2002.

[62] Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. 2008.

[63] Matthew Marge, Satanjeev Banerjee, and Alexander I. Rudnicky. Using the Amazon Mechanical Turk for transcription of spoken language. In *Proceedings of ICASSP*, pages 5270–5273. IEEE, 2010.

[64] Richard McCreadie, Craig Macdonald, and Iadh Ounis. Crowdsourcing Blog Track Top News Judgments at TREC. In Matthew Lease, Vitor Carvalho, and Emine Yilmaz, editors, *Proceedings of the Workshop on Crowdsourcing for Search and Data Mining at the WSDM*, pages 23–26, Hong Kong, China, February 2011.

[65] Ian McGraw, James R. Glass, and Stephanie Seneff. Growing a Spoken Language Interface on Amazon Mechanical Turk. In *Proceedings of INTERSPEECH*, pages 3057–3060, 2011.

[66] Ian McGraw, Alexander Gruenstein, and Andrew M. Sutherland. A Self-labeling Speech Corpus: Collecting Spoken Words with an Online Educational Game. In *Proceedings of INTERSPEECH*, pages 3031–3034, 2009.

[67] P. McNamee, M. Dredze, A. Gerber, N. Garera, T. Finin, J. Mayfield, C. Piatko, D. Rao, D. Yarowsky, and M. Dreyer. HLTCOE Approaches to Knowledge Base Population at TAC 2009. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[68] Paul McNamee, James Mayfield, Douglas W. Oard, Tan Xu, Ke Wu, Veselin Stoyanov, and David Doermann. Cross-Language Entity Linking in Maryland during a Hurricane. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[69] Rada Mihalcea, Courtney Corley, and Carlo Strapparava. Corpus-based and Knowledge-based Measures of Text Semantic Similarity. In *Proceedings of AAAI*, pages 775–780, 2006.

[70] David Milne and Ian H. Witten. Learning to Link with Wikipedia. In *Proceeding of*

the 17th ACM conference on Information and knowledge mining - CIKM '08, page 509, New York, New York, USA, 2008. ACM Press.

[71] Saeedeh Momtazi, Sanjeev Khudanpur, and Dietrich Klakow. A Comparative Study of Word Co-occurrence for Term Clustering in Language Model-based Sentence Retrieval. In *HLT: 2010 NAACL*, pages 325–328, Los Angeles, California, 2010. Association for Computational Linguistics.

[72] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and Arnold Jung. Cross-Lingual Cross-Document Coreference with Entity Linking. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[73] Joanna Mrozinski, Edward Whittaker, and Sadaoki Furui. Collecting a Why-Question Corpus for Development and Evaluation of an Automatic QA-system. In *Proceedings of ACL-08: HLT*, pages 443–451, Columbus, Ohio, USA, 2008.

[74] Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. Divide and Conquer: Crowdsourcing the Creation of Cross-Lingual Textual Entailment Corpora. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 670–679, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.

[75] Xiaochuan Ni, J.T. Sun, Jian Hu, and Zheng Chen. Cross Lingual Text Classification By Mining Multilingual Topics From Wikipedia. In *Proceedings of the fourth WSDM*, number 49, pages 375–384. ACM, 2011.

[76] Scott Novotney and Chris Callison-Burch. Cheap, Fast and Good Enough: Automatic Speech Recognition with Non-Expert Transcription. In *Proceedings of HLT:NAACL*, pages 207–215, Los Angeles, California, June 2010. Association for Computational Linguistics.

[77] Carol Peters, Julio Gonzalo, Martin Braschler, and Michael Kluck. Comparative Evaluation of Multilingual Information Access Systems, 4th Workshop of the Cross-Language Evaluation Forum, CLEF 2003. In *CLEF*, volume 3237 of *Lecture Notes*

in Computer Science. Springer, 2004.

[78] Danuta Ploch, Leonhard Hennig, Ernesto William De Luca, and Sahin Albayrak. DAI Approaches to the TAC-KBP 2011 Entity Linking Task. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[79] Will Radford, Ben Hachey, Matthew Honnibal, Joel Nothman, and James R. Curran. Naive but Effective NIL Clustering Baselines – CMCRC at TAC2011. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[80] Will Radford, Ben Hachey, Joel Nothman, Matthew Honnibal, and James R. Curran. Document-level Entity Linking: CMCRC at TAC 2010. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[81] Philip Resnik. Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In *In Proceedings of the 14th IJCAI*, pages 448–453, 1995.

[82] Bryan C. Russell, Antonio Torralba, Kevin P. Murphy, and William T. Freeman. LabelMe: A Database and Web-Based Tool for Image Annotation. *International Journal of Computer Vision*, 77(1-3):157–173, May 2008.

[83] Mehran Sahami and Timothy D. Heilman. A Web-based Kernel Function for Measuring the Similarity of Short Text Snippets. In *WWW '06*, pages 377–386, New York, NY, USA, 2006. ACM.

[84] Celina Santamaría, Julio Gonzalo, and Javier Artiles. Wikipedia as Sense Inventory to Improve Diversity in Web Search Results. In *Proceedings of ACL*, number July, pages 1357–1366. Association for Computational Linguistics, 2010.

[85] Dan Shen, Michael Wiegand, Andreas Merkel, Stefan Kazalski, Sabine Hunsicker, Jochen L. Leidner, and Dietrich Klakow. The Alyssa System at TREC QA 2007: Do We Need Blog06? In *Proceedings of the 16th Text Retrieval Conference (TREC)*, Gaithersburg, MD, USA, 2007.

[86] M. Six Silberman, Joel Ross, Lilly Irani, and Bill Tomlinson. Sellers' Problems in

Human Computation Markets. In *Proceedings of the ACM SIGKDD Workshop on Human Computation*, HCOMP '10, pages 18–21, New York, NY, USA, 2010. ACM.

[87] Alex J. Smola, Bernhard Schölkopf, and Bernhard Sch Olkopf. A Tutorial on Support Vector Regression. Technical report, Statistics and Computing, 2003.

[88] Rion Snow, Brendan O Connor, Daniel Jurafsky, and Andrew Y Ng. Cheap and Fast – But is it Good? Evaluating Non-Expert Annotations for Natural Language Tasks. In *Proceedings of ACL-08:HIT*, 2008.

[89] Alexander Sorokin and David Forsyth. Utility Data Annotation with Amazon Mechanical Turk. volume 0, pages 1–8, Los Alamitos, CA, USA, 2008. IEEE Computer Society.

[90] Harish Srinivasan, John Chen, and Rohini Srihari. Cross Document Person Name Disambiguation Using Entity Profiles. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[91] Mihai Surdeanu, Massimiliano Ciaramita, and Hugo Zaragoza. Learning to Rank Answers on Large Online QA Collections. In *Proceedings of ACL-08: HLT*, pages 719–727, 2008.

[92] Ling-Xiang Tang, Shlomo Geva, Andrew Trotman, Yue Xu, and Kelly Itakura. Overview of the NTCIR-9 Crosslink Task : Cross-Lingual Link Discovery. In Noriko Kando, Daisuke Ishikawa, and Miho Sugimoto, editors, *9th NTCIR Workshop Meeting on Evaluation of Information Access Technologies: Information Retreival, Question Answering and Cross-Lingual Information Access*, pages 437–463, National Center of Sciences, Tokyo, 2011. National Institute of Informatics.

[93] Stefanie Tellex, Boris Katz, Jimmy Lin, Aaron Fernandes, and Gregory Marton. Quantitative Evaluation of Passage Retrieval Algorithms for Question Answering. In *Proceedings of SIGIR*, number July, 2003.

[94] Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. Feature-

Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In *Proceedings of the NAACL*, pages 173–180. Association for Computational Linguistics, 2003.

[95] Howard Turtle and William B. Croft. Inference Networks for Document Retrieval. In *Proceedings of SIGIR '90*, pages 1–24, New York, NY, USA, 1990. ACM.

[96] Vladimir Vapnik. *The Nature of Statistical Learning Theory*. Statistics for Engineering and Information Science, Springer, 2000.

[97] Vasudeva Varma, Vijay Bharat, Sudheer Kovelamudi, Praveen Bysani, Santosh GSK, Kiran Kumar N, Kranthi Reddy, Karuna Kumar, and Nitin Maganti. IIIT Hyderabad at TAC 2009. In *Proceedings of Text Analytics Conference (TAC2009)*, 2009.

[98] Vasudeva Varma, Praveen Bysani, Kranthi Reddy, Vijay B. Reddy, Sudheer Kovelamudi, Srikanth Reddy Vaddepally, Radheshyam Nanduri, Kiran Kumar N, Santhosh Gsk, and Prasad Pingali. IIIT Hyderabad in Guided Summarization and Knowledge Base Population. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[99] Suzan Verberne, Hans van Halteren, Daphne Theijssen, Stephan Raaijmakers, and Lou Boves. Learning to Rank QA Data. In *SIGIR Workshop on Learning to Ranking*, 2009.

[100] Ellen Voorhees and Donna Harman. TREC Experiment and Evaluation in Information Retrieval. MIT Press, MA, USA, 2005.

[101] Ellen M. Voorhees. Overview of the TREC 2001 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2001*, pages 42–51, 2001.

[102] Ellen M. Voorhees. Overview of the TREC 2002 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2002*, pages 115–123, 2002.

[103] Ellen M. Voorhees. Overview of the TREC 2003 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2003*, pages 54–68, 2003.

[104] Ellen M. Voorhees. Overview of the TREC 2004 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2004*, 2004.

[105] Ellen M. Voorhees and Hoa Trang Dang. Overview of the TREC 2005 Question Answering Track Evaluation. In *Text Retrieval Conference TREC 2005*, 2005.

[106] Ellen M. Voorhees and Dawn M. Tice. Overview of the TREC-8 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-8*, pages 83–105, 1999.

[107] Ellen M. Voorhees and Dawn M. Tice. Overview of the TREC-9 Question Answering Track Evaluation. In *Text Retrieval Conference TREC-9*, pages 71–80, 2000.

[108] Jacob Whitehill, Paul Ruvolo, Tingfan Wu, Jacob Bergsma, and Javier Movellan. Whose Vote Should Count More: Optimal Integration of Labels from Labelers of Unknown Expertise. In *Proceedings of NIPS*, number 1, pages 1–9, 2009.

[109] Michael Wiegand, Saeedeh Momtazi Stefan Kazalski, Fang Xu, Grzegorz Chrupała, and Dietrich Klakow. The Alyssa System at TAC QA 2008. In *Proceedings of TAC 2008*, 2008.

[110] William Addison Woods. Progress in Natural Language Understanding: an Application to Lunar Geology. In *Proceedings of National Computer Conference and Exposition*, AFIPS '73, pages 441–450, New York, NY, USA, 1973. ACM.

[111] Xiaofeng Wu, Junhui Li, Jie Jiang, Yifan He, and Andy Way. DUC Multi-Engine MT System for CWMT'2011. In *Proceedings of China Workshop on Machine Translation CWMT'2011*, 2011.

[112] Youzheng Wu, Jun Zhao, Bo Xu, and Hao Yu. Chinese Named Eentity Recognition Based on Multiple Features. In *Proceedings of EMNLP: HLT '05*, pages 427–434, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics.

[113] Zhibiao Wu and Martha Palmer. Verbs semantics and lexical selection. In *Proceedings of ACL '94*, pages 133–138. Association for Computational Linguistics, 1994.

[114] Fen Xia, Tie-Yan Liu, Jue Wang, Wensheng Zhang, and Hang Li. Listwise Approach to Learning to Rank: Theory and Algorithm. In *Proceedings of ICML08*, pages

1192–1199, 2008.

[115] Fang Xu, Stefan Kazalski, Grzegorz Chrupala, Benjamin Roth, Xujian Zhao, Michael Wiegand, and Dietrich Klakow. Saarland University Spoken Language Systems Group at TAC KBP 2011. In *Proceedings of TAC 2011*, 2011.

[116] Fang Xu and Dietrich Klakow. Paragraph Acquisition and Selection for List Question Using Amazon's Mechanical Turk. In *Proceedings of LREC'10*, May 2010.

[117] Fang Xu and Dietrich Klakow. Proximity-based Passage Ranking for Question Answering. In *12th Dutch-Belgian Information Retrieval Workshop*, 2012.

[118] Gae-won You, Seung-won Hwang, Young-In Song, Long Jiang, and Zaiqing Nie. Mining Name Translations from Entity Graph Mapping. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 430–439, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[119] Yisong Yue, Thomas Finley, Filip Radlinski, and Thorsten Joachims. A Support Vector Method for Optimizing Average Precision. In *Proceedings of SIGIR*, 2007.

[120] Omar F. Zaidan and Chris Callison-Burch. Crowdsourcing Translation: Professional Quality from Non-Professionals. In *Proceedings of the 49th ACL:HLT*, pages 1220–1229, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[121] Chengxiang Zhai and John Lafferty. A Study of Smoothing Methods for Language Models Applied to Information Retrieval. *ACM Trans. Inf. Syst.*, 22(2):179–214, April 2004.

[122] Tao Zhang, Kang Liu, and Jun Zhao. NLPR TAC Entity Linking System at TAC2011. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[123] Wei Zhang, Chew Lim Tan, Yan Chuan Sim, and Jian Su. NUS-I2R: Learning a Combined System for Entity Linking. In *Proceedings of Text Analytics Conference (TAC2010)*, 2010.

[124] Wei Zhang, Chew Lim Tan, Jian Su, Bin Chen, Wenting Wang, Zhiqiang Toh, Yanchuan Sim, Yunbo Cao, and Chin Yew Lin. I2R-NUS-MSRA at TAC 2011: Entity Linking. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[125] Ying Zhao and George Karypis. Evaluation of Hierarchical Clustering Algorithms for Document Datasets. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 515–524. ACM Press, 2002.

[126] Yu Zhao, Weipeng He, Zhiyuan Liu, and Maosong Sun. THUNLP at TAC KBP 2011 in Entity Linking. In *Proceedings of Text Analytics Conference (TAC2011)*, 2011.

[127] Jing Zheng, Necip Fazil Ayan, Wen Wang, and David Burkett. Using Syntax in Large-Scale Audio Document Translation. In *INTERSPEECH*, pages 440–443, 2009.