NAMED ENTITY DISAMBIGUATION FROM WEB TEXT

XU JIAN

Ph.D

The Hong Kong Polytechnic University

2014

The Hong Kong Polytechnic University

Department of Computing

Named Entity Disambiguation

from Web Text

XU Jian

A Thesis Submitted in Partial Fulfillment of the

Requirements for the Degree of Doctor of Philosophy

March 2014

CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

________________________________(Signed)

__________XU Jian______________(Name of Student)

Abstract

Named entity disambiguation is the problem of grouping **name mentions** into clusters, with each cluster referring to the same underlying **entity**. In this thesis, we focus on named entity disambiguation from web text, because finding information about person on the Internet is one of the most common activities of online users. Person's names, however, are highly ambiguous with a large number of people sharing the same name. Named entity disambiguation therefore becomes increasingly important for many applications such as information retrieval, question answering, cross-document co-reference, relation discovery and so on. This leads to our study of named entity disambiguation over the Internet. In general, named entity disambiguation for web text includes two tasks: (1) Web Person Disambiguation (WPD), which groups search results into different clusters with each cluster referring to the same person; and (2) personal profile extraction (PPE), which can help build each person's relational information in the cluster.

The main challenges in named entity disambiguation include (1) how to select meaningful features that are unique to identify named entities; (2) how to guarantee high performance in WPD, even if there is no prior knowledge of the number of persons having the same name; (3) how to obtain and select quality training data from an external knowledge base for personal profile extraction (PPE), since manually annotated data is costly to yield and limited in scale.

In this thesis, we explore the use of more semantically relevant information for named entity disambiguation on web text. For WPD, our supervised approach can make good use of naturally annotated resource, Wikipedia in particular to alleviate manual annotation efforts and domain dependence problems. We also investigate the usage of keywords as semantically more meaningful information units for WPD. Based on meaningful keyword features, we investigate a hierarchical co-reference resolution technique to place ambiguous person names into different clusters. Our disambiguation method does not require a predefined number of persons and can produce good quality clusters for each person. For PPE, we build a personalized profile by identifying relational facts. Our approach is to incorporate two semantic constraints, including both trigger word and entity type which can help reduce noisy data in profile extraction. Both WPD and PPE are built within the framework of graphical models, which can provide sequential structure for semantic feature extraction and tree structure for both name disambiguation and profile extraction. The methods in this thesis are evaluated on publicly available datasets so that performance comparisons can be made to state-of-the-art works and our approach is proven to be effective in named entity disambiguation.

List of Publications

1. **Xu**, **J.**, Lu, Q. 2013. PolyUCOMP-CORE_TYPED: Computing Semantic Textual Similarity using Overlapped Senses. In *Second Joint Conference on Lexical and Computational Semantics (*SEM) Volume 1: In Proceedings of the Main Conference and the Shared Task: Semantic Textual Similarity*, pages 90-95, Atlanta, Georgia.

2. **Xu**, **J.**, Lu, Q., Liu, Z. 2012. Combining Classification with Clustering for Web Person Disambiguation. In *Proceedings of WWW2012, Companion,* pages 637-638, Lyon, France.

3. **Xu**, **J.**, Lu, Q., Liu, Z., Chai, J. 2012. Topic Sequence Kernel. In *The Eighth Asia Information Retrieval Societies Conference (AIRS2012)*, pages 457-466, Tianjin, China.

4. **Xu**, **J.**, Lu, Q., Liu, J., Xu, R. 2012. NLPComp in TAC 2012 Entity Linking and Slot Filling. In *Proceedings of Text Analytics Conference (TAC2012)*, Gaithersburg, Maryland, USA.

5. **Xu**, **J.**, Lu, Q., Liu, Z. 2012. PolyUCOMP: Combining Semantic Vectors with Skip bigrams for Semantic Textual Similarity. In *First Joint Conference on Lexical and Computational Semantics (*SEM),* pages 524–528, Montreal, Canada.

6. Liu, J., Xu, R., Lu, Q., **Xu**, **J**. 2012. Explore Chinese Encyclopedic Knowledge to Disambiguate Person Names. In *Proceedings of the Second CIPS-SIGHAN Joint*

Conference on Chinese Language Processing, pages 138-145, Tianjin, China.

7. **Xu**, **J**., Liu, Z., Lu, Q., Liu, P., Wang, C. 2011. PolyUCOMP in TAC 2011 Entity Linking and Slot Filling. In *Proceedings of Text Analytics Conference (TAC2011)*, Gaithersburg, Maryland, USA., Electronic Copy at TAC2011 Site: http://www.nist.gov/tac/publications/2011/participant.papers/PolyUCOMP.proceedings.pdf

8. Liu, Z., Lu, Q., **Xu, J**. 2011. High Performance Clustering for Web Person Name Disambiguation using Topic Capturing. In *Proceedings of the International Workshop on Entity-Oriented Search (EOS)*, pages 1-6, Beijing, China.
http://research.microsoft.com/en-us/um/beijing/events/eos2011/9.pdf

Acknowledgements

I wish to express my gratitude to everyone who contributed to the completion of this doctoral dissertation.

First and foremost, I would like to show my deepest thanks to my supervisor Prof. Qin Lu. Over the past three years, she has given me full encouragement and support in my research work. She has enlightened me with her deep insights in the current research domain. I have benefited a lot from her guidance in domain knowledge acquisition, algorithm design, experimental setup and thesis writing.

Secondly, I would like to thank Dr. Wenjie Li for her valuable advices in information retrieval. It is also my great pleasure to thank the colleagues in the Chinese Computing and Natural Language Processing Lab for their constant helps through the years. I thank you all: Mr. Zhengzhong Liu, Ms. Dan Xiong, Ms. Chengyao Chen, Dr. Dehong Gao, Dr. Renxian Zhang, and Dr. Junyi Chai. Sincere thanks also go to my thesis committee members Prof. James Liu, Prof. Irvin King and Prof. Yiuming Cheung for their kind review and feedback.

I must thank my wife and my family for their constant support and great encouragement over the years without which I would never be able to realize my dream.

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

Searching information about persons on the Internet is one of the most common activities for Internet users, with an estimated 30% web queries containing person names (Artiles et al., 2005, 2007). In a survey of Google 2013 search trends, 3 out of the top 10 searches are person names. It is interesting to note that statistics from the 1990 U.S. Census bureau show that only 90, 000 different names are shared by 100 million people (Artiles et al., 2005). However, due to the ambiguity of person names (for example, the name mentioned "*Michael Jordan*" can be referred to the American basketball player as well as the computer science professor at UC Berkeley), a literal name as a lexical sequence can appear over the Internet in large quantity. Thus, many returned Web pages containing person names may not refer to the same person. Person names on the Web pages are, therefore, highly ambiguous. Furthermore, celebrity or popular names tend to monopolize search results as most of current search engines will always return the most highly cited persons. Hence, users would be inundated by a vast amount of unwanted information and would need to add more query items in order to locate the target Web pages.

Generally speaking, named entity disambiguation groups **name mentions** into clusters, with each cluster referring to the same underlying **entity**. A **name mention** is defined as an observed lexical sequence for a named entity in a text and an **entity** is a specific, disambiguated individual. For instance, "*Bill Clinton*" and "*Clinton*" are mentions for the entity *Bill Clinton*, the former U.S. president.

In this work, we focus on studying named entity disambiguation over the Internet. In general, named entity disambiguation for web text includes two tasks: (1) **Web Person Disambiguation** (WPD) which groups search results into different clusters, with each cluster referring to the same person; and (2) **Personal Profile Extraction** (PPE) which aims to build each person's relational information within the cluster. Ideally, when a user searches for a person's name, the search engine should return separate groups of documents and each group refers to the same individual, possibly with a list of relational facts as his/her personal profile (Artiles et al., 2007, 2009, 2010). To reach this goal, however, it is important to disambiguate person names by grouping documents into different clusters. Henceforth, in each cluster of a disambiguated person, relational facts (birth date, children, sibling, education etc.) can be extracted to build up a personalized profile. Optionally, to facilitate knowledge population, disambiguated persons with profiles can be linked to named entities in an existing knowledge base[1] (Lehmann et al., 2010; McNamee et al., 2010; Ji et al., 2011).

To conduct WPD, most current research works use clustering methods, because the number of persons as unique real-world entities is not known beforehand. To utilize clustering methods, there are two main challenges. The first is related to feature selection, and the second is how to select appropriate clustering methods. Early works on feature selection attempt to use a combination of different features such as tokens in texts, URLs or titles, n-gram features, snippets and so on (Bagga and Baldwin, 1998; Chen et al., 2009; Long and Shi, 2010). However, these feature selection approaches face two problems. Firstly, simple features can be more efficient, but may suffer from a data sparseness issue. However, more features could also introduce more noise which can degrade system performance. Secondly, most of

[1] A knowledge base is a database containing information about entities, their attributes and relationships. For example, Wikipedia can be viewed as a knowledge base about people, organizations, events and so on.

these features are extracted using unsupervised methods. Few researchers investigated supervised approaches to extract WPD features, because it is costly to obtain manually annotated training data and most existing corpora are confined to a particular domain and small in size. To take advantage of supervised learning methods and at the same time not drain resources for manually annotated data, we explore the use of naturally annotated resources, Wikipedia in particular, to automatically obtain a large corpus of annotated sentences, which can then be used to train an appropriate learning model. When choosing features for a supervised WPD method, keywords can provide more semantically relevant information units for a person, for example, keywords “basketball player” and “computer scientist” can help separate two persons having the same name, for example “*Michael Jordan*”.

For clustering algorithms, researchers have tried such approaches as the hierarchical agglomerative clustering (HAC) (Elmacioglu et al., 2007; Chen et al., 2009; Long and Shi, 2010), K-Means clustering (Rao et al., 2007) and fuzzy ant clustering (Lefever et al., 2009). However, these approaches require tuning a threshold to find the number of clusters, which is a particular challenging problem of WPD because we do not know beforehand how many clusters exist in the search results for a given person’s name. In this thesis, we attempt to solve this problem by using the hierarchical co-reference resolution technique, which recursively partitions entities into a tree structure with latent sub-entities as child nodes and person names as observable leaf nodes. Person names are then disambiguated by deciding whether two entity nodes are co-referential or not. The benefit of our approach lies in that it does not need to tune the termination threshold to determine the number of clusters and has a greater scalability due to its hierarchical organization of person names.

Recent works on PPE favor the distant supervision approach because it can employ an external knowledge base (Freebase, for example) as a source for semi-supervised learning (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2010; Hoffmann et al., 2011; Sun et al., 2011). This approach can reduce the manual efforts of data annotation, but selecting quality training data in the knowledge base becomes a very important issue. To address this issue, we incorporate two semantic constraints, including both trigger word and entity type, into a graphical model. We observe that the relation between entities can be triggered by lexical words such as "*son/daughter*" which are indicative of *person.parents_children* relation if they co-occur with two entities in a sentence; and the two entities must be confined to certain entity types such as the entity type *Persons* in this relation. Entity types control the selectional preference of two entities that participate in a relation. Trigger words add more positive evidences that are closely related to the target relations, which can help reduce the use of noisy data in profile extraction.

Both WPD and PPE are built within the framework of graphical models, which can provide sequential structure for semantic feature extraction and tree structure for both name disambiguation and profile extraction. Graphical models can capture the contextual clues for extracting semantic features in the sequential structure, and reduce the quadratic number of decisions when resolving name mention ambiguity in the hierarchical tree. Furthermore, semantic constraints for personal profiling can be directly incorporated into the graphical models. The methods in this thesis are evaluated on publicly available datasets so that performance comparisons can be made to state-of-the-art works and our approach is proven to be effective in named entity disambiguation.

The main contributions of this thesis can be categorized into three aspects:

(1) Semantic keyword extraction using naturally annotated resources. We

mainly explore the use of keywords as semantic features for Web Person Disambiguation (WPD). The supervised keyword extractor is trained on a large corpus automatically generated from naturally annotated resources, Wikipedia in particular. Our keyword extractor requires no manual annotation efforts and is much less domain sensitive due to Wikipedia’s wide coverage.

(2) Web Person Disambiguation using a hierarchical co-reference technique. We investigate the hierarchical co-reference model to disambiguate person names. This model does not need to manually tune the number of clusters, and can incorporate many person-specific features.

(3) Personal profile extraction using semantic constraints. To reduce the noise in training data, we investigate the usage of two semantic constraints in a graphical model to extract relational facts in a profile. Trigger words add positive evidences that are closely related to the target relations and entity types control the selectional preferences of two entities in a relation.

The remainder of this thesis is organized as follows:

Chapter 2 gives a literature overview of feature extraction, Web Person Disambiguation (WPD) and Personal Profile Extraction (PPE). **Chapter 3** gives details on graphical models and parameter learning. **Chapter 4** presents the automatic extraction of keywords using naturally annotated resources. **Chapter 5** presents contextual relevance weighting of features and the hierarchical co-reference technique for WPD. **Chapter 6** introduces personal profile extraction using semantic constraints. **Chapter 7** introduces named entity linking (NEL). **Chapter 8** concludes the studies in this thesis and identifies directions for future work.

Chapter 2 Literature Review

Named entity disambiguation has attracted a great deal of attention from researchers, due to its wide application in information retrieval (Pantel and Fuxman, 2011), author disambiguation (Culotta et al., 2007; Wick et al., 2013), knowledge base population (Dredze et al., 2010) and so on. In this thesis, we focus on named entity disambiguation from web text, since finding information about people on the Internet is one of the most common activities of online users. In general, named entity disambiguation for web text includes two tasks: (1) Web Person Disambiguation (WPD) which groups search results into different clusters with each cluster referring to the same person; and (2) personal profile extraction (PPE) which can help build each person's relational information in the cluster. For WPD, semantically relevant features need to be extracted to identify a person. For this purpose, we explore the use of keywords as semantic features. Hence, we review previous researchers' works on feature extraction (**Keyword Extraction**, KWE in short), Web Person Disambiguation (WPD) and Personal Profile Extraction (PPE), respectively.

2.1 Keyword Extraction

Keywords give a semantic summarization of documents and are used in text clustering (Hammouda et al., 2005), text categorization (Hulth and Megyesi, 2006) summarization (Litvak and Last, 2008), and web person disambiguation (Xu et al., 2012). They can also be used for terminology extraction and domain-specific dictionary building (Mihalcea and Tarau, 2004). To extract keywords automatically, numerous approaches have been proposed and can be roughly divided into three categories: simple statistics method, graph-based ranking methods and supervised machine learning methods.

2.1.1 Simple Statistics Methods for KWE

Simple statistics like term frequency, word co-occurrence, χ^2 measure and term informativeness are used to assess the affinity between two consecutive words (Matsuo and Ishizuka, 2004; Wu and Giles, 2013). Matsuo and Ishizuka (2004) extracted keywords from a single document without using an external corpus. They initially found the most frequent single words in a document and then calculated co-occurrence statistics between a candidate word and frequent words. Keywords are selected if they have a high χ^2 score. This approach has an advantage in that it is simply based on a single document, but it can be restricted by low frequency terms since low frequency terms can be keywords as well. To utilize external resources for keyword extraction, Wu and Giles (2013) created a context-aware term informativeness model using semantic similarity/relatedness between a term's context and a term's featured context with highest authority scores in a knowledge base. They collected the term’s featured context from the Web or a knowledge base, Wikipedia for example. They then measured the term’s informativeness by computing semantic relatedness between the term’s context and its featured context using a Wikipedia-based ESA approach (Gabrilovich and Markovitch, 2007). Finally, they chose the term as a keyword if it has a high informativness score. Their methods require no training procedure, but need a large corpus in order to compute the semantic relatedness between two contexts. In addition, they have limited the length of a keyword to 4.

2.1.2 Graph-based Ranking Methods for KWE

A graph-based ranking approach for keyword extraction is one way to determine a term’s importance by mutual “voting” strategy in a graph. The graph can be constructed from a text using words as nodes and links will be added into the graph if two words are adjacent to one another or co-occur within a sentence or paragraph. Zha (2002) applied the mutual reinforcement principle to extract keywords from a sentence and ranked keywords using the HITS algorithm. In the TextRank model, Mihalcea and Tarau (2004) first built a graph of lexical units, and an edge between two lexical units will be added into this graph if they co-occur within a specified window size. They then ranked lexical units on this graph using the PageRank algorithm. The sequences of adjacent top-ranked words are treated as keywords. The TextRank model builds a graph on a single document by assuming that documents are independent from each other. To investigate the interactions between documents, Wan and Xiao (2008) proposed the CollabRank approach which first grouped documents into different clusters. They then built a graph using words from different documents within each cluster and ranked words on the graph using the PageRank algorithm. In so doing, they obtained word scores on different clusters. For each document, the weight of a keyword sums up its word scores across different clusters.

2.1.3 Supervised Methods for KWE

Unlike the simple statistics method and graph-based ranking approach, supervised approaches for keyword extraction can obtain a better performance if the manually annotated training data fits the target domain. In supervised learning for keyword extraction, candidate keywords are extracted first and classified as true keywords on the basis of different features, such as TFIDF scores (frequency of a candidate keyword in a document and its frequency in a corpus), distance values (the number of words that precede the candidate keyword's first appearance), syntactic features (part-of-speech tags, NP-chunks). Turney (1999) first used the genetic algorithm and bagged C4.5 decision tree for keyword extraction. Frank et al. (1999) and Witten et al. (2000) used the Naïve Bayes classifier to extract keywords. As a departure from using the manually annotated training data, Xu et al. (2012) proposed to use the anchor texts in Wikipedia articles for keyword extraction by means of the Naïve Bayes classifier. In addition to the Naive Bayes classifier, Hulth (2003) combined a number of classifiers in Bagging (Breiman, 1996) for keyword extraction from short texts of scientific abstracts with lexical and syntactic features. Lopez and Romary (2010) trained decision tree (C4.5), multi-layer perceptron and Support Vector Machines (SVMs) using structural features (title, abstract, introduction etc.), content features (phraseness, informativeness), and lexical/semantic features. Zhang et al. (2008) applied the Conditional Random Fields (CRFs) model for Chinese keyword extraction and they reported that the CRFs model outperforms the SVMs, multiple linear regression (MLR), logistic regression (Logit) and two baselines which simply use TFIDF and distance scores in their experiments.

2.1.4 Summary for KWE

Simple statistics methods for keyword extraction require no training, but they can be restricted by low frequency terms because low frequency terms can be keywords as well. Graph-based ranking approaches fully utilize the "voting" strategy in a graph, but their performance can be limited by the length/number of sentences or paragraphs since short sentences or paragraphs are not sufficient to build a graph to rank candidate keywords. Supervised methods often show their superiority in keyword extraction if the manually annotated training data is provided. However, the manually annotated training data is costly to yield. Also, it can be domain sensitive, thus limiting keyword extraction to a small scale. Worse still, previous supervised methods (Naive Bayes or decision tree classifier) put limits on the length of keywords and can detect these keywords simply containing 2-4 tokens.

2.2 Web Person Disambiguation

As one of the named entity disambiguation tasks, Web Person Disambiguation (WPD) targets at grouping search results into different clusters with each cluster referring to the same person (Artiles et al., 2009, 2010). It is a challenging task since an entity (such as "*Michael Jordan*", the American basketball player) can be described by multiple name mentions (e.g., "*Michael Jeffrey Jordan*", "*MJ*", "*Jordan*", "*Air Jordan*" and "*His Airness*") and a name mention (e.g., "*Michael Jordan*") can refer to multiple entities (the American basketball player or the computer science professor at UC Berkeley). To resolve ambiguous person names, most current researchers have tried such approaches as the hierarchical agglomerative clustering (HAC) (Elmacioglu et al., 2007; Chen et al., 2009; Long and Shi, 2010), K-Means clustering (Rao et al., 2007) and Fuzzy Ant Clustering (Lefever et al., 2009). In addition, classification methods have also been applied in WPD (Lefever et al., 2007; Han and Zhao, 2009).

2.2.1 Clustering Methods for WPD

Clustering is to group documents into different clusters. Currently in WPD, Hierarchical Agglomerative Clustering (HAC), K-Means Clustering, and Fuzzy Ant Clustering are used.

The HAC algorithm treats each document as a singleton cluster at the start, and then merges pairs of clusters using different linkage metrics until all clusters are agglomerated into one cluster which contains all documents. In the WPD task, the HAC algorithm is widely used because it can determine the number of clusters by manually assigning a threshold (Elmacioglu et al., 2007; Balog et al., 2009; Gong and Oard, 2009; Ikeda et al., 2009; Long and Shi, 2010). Balog et al. (2009) used three kinds of clustering approaches to generate different clustering solutions and then applied a voting strategy to combine them. Gong and Oard (2009) explored local and global features to find the thresholds for the HAC algorithm. Ikeda et al. (2009) used a two-stage clustering approach to disambiguate person names. They grouped documents into clusters using the HAC algorithm in the first step, and then refined clusters by finding the compound keywords in the cluster with the maximum number of documents. Other clusters will be merged into this cluster if they share the compound keywords.

In addition to the HAC algorithm, the K-Means clustering algorithm has also been applied to WPD (Kozareva et al., 2007; Rao et al., 2007). Rao et al. (2007) attempted to determine cluster number *K* based on the various proportions of documents. In contrast to the HAC and K-Means algorithms, Fuzzy Ant Clustering does not rely on prior knowledge of the number of clusters. Each document is placed into a group by its membership in the interval [0,1]. Lefever et al. (2009) and

Venkateshan (2009) used the Fuzzy Ant Clustering algorithm in WPD. However, this algorithm has its own limitations due to the uneven distribution of data points and large variations in cluster size.

2.2.2 Classification Methods for WPD

In addition to HAC, K-Means, and Fuzzy Ant Clustering approaches, other approaches have also been studied, for example, combining classification with clustering (Lefever et al., 2007), semi-supervised clustering (Sugiyama and Okumura, 2007), graph-based clustering (Iria et al., 2007; Smirnova et al., 2010), KNN classifier (Han and Zhao, 2009). Lefever et al. (2007) used the Ripper rule learner to obtain a set of classification rules from the training and trial data and applied these rules to generate “seed” clusters for the next-step clustering algorithm. Sugiyama and Okumura (2007) adopted a semi-supervised clustering approach based on labeled documents collected from Wikipedia and the Internet. They then merged the testing documents into the labeled documents if they are similar. In terms of graph-based clustering, Smirnova et al. (2010) first found the topically related pages for a particular person page using the random walk approach and then employed the HAC algorithm to group documents with no link preference in the first stage. Han and Zhao (2009) tried the KNN classifier based on the professional categories that are extracted from the Freebase. The documents classified into the same professional category will be grouped into one cluster.

2.2.3 Summary for WPD

In sections 2.2.1 and 2.2.2, we present the general methods in WPD: Hierarchical Agglomerative Clustering (HAC), K-Means Clustering, and Fuzzy Ant

Clustering. Meanwhile, we quickly review many other WPD methods, including semi-supervised clustering, graph-based clustering and KNN classifier.

However, the clustering approaches for WPD require tuning a threshold to find the number of clusters, which is a particularly challenging problem of WPD because we do not know beforehand how many clusters exist in the search results for a given person’s name. Additionally, the clustering algorithm (HAC for example) is rather sensitive to the manually tuned threshold. A minor change in threshold would cause a great fluctuation in clustering performance. This problem is complicated when the number of clusters varies from name to name. Classification methods for WPD require manually annotated training data, which is impractical since there is no sure way to prepare all possible training instances for future testing data. Moreover, classification based on professions would fail if a person has more than one profession.

2.3 Personal Profile Extraction

As the second task in the named entity disambiguation framework, Personal Profile Extraction (PPE) aims to extract relational facts from the clusters created by the WPD module, such as *spouse*, *birthplace*, *siblings*, *parents* and so on. Hence, PPE can be seen as the task of predicting relation types between two entities. Different tasks have been set up to extract relations between named entities (persons, locations, organizations), including Message Understanding Conference (MUC) (Grishman and Sundheim, 1996), Automatic Content Extraction (ACE) (Doddington et al., 2004), slot filling task which is to learn missed attributes for entities in Knowledge Base Population of Text Analysis Conference (TAC KBP) (Dang and Owczarzak, 2009; Ji et al., 2010; Ji et al., 2011). To predict relation types, numerous approaches have been proposed. In general, they are roughly divided into two categories: rule-based method and machine learning method.

2.3.1 Rule-based Methods for PPE

Rule-based methods simply take a set of seed examples or hand-written extraction patterns for PPE, for example, DIPRE (Brin, 1998), Snowball (Agichtein and Gravano, 2000), KnowItAll (Etzioni et al., 2005), TextRunner (Banko et al., 2007). Some researchers have used part-of-speech patterns, entity type plus verb patterns or trigger words (Byrne et al., 2010; Varma et al., 2010; Xu et al., 2011).

DIPRE extracts the relation of *author_book* with a small set of seed example pairs (Brin, 1998). Similar to DIPRE, the Snowball system extracts the relation of organization, location with a set of seed tuples (Agichtein and Gravano, 2000). Snowball differs from DIPRE in that it weighs features for relation pairs instead of using exact feature matching. Unlike DIPRE and Snowball, KnowItAll automates the extraction of entity facts from the Web in a large scale using a set of eight domain-independent patterns (Etzioni et al., 2005). The patterns in KnowItAll are based on a Noun Phrase chunker. DIPRE, Snowball, KnowItAll use manually prepared relations for training purposes. On a separate note, TextRunner, a relation extraction system, automatically discovers open relations from text in a self-supervised manner using syntactic patterns (Banko et al., 2007). It starts with a small corpus sample and extracts triples representing binary relation from sentences in the corpus. TextRunner then automatically labels the extracted triples as positive and negative training data with several heuristic constraints.

Byrne et al. (2010) used the canonical form of the entity + verb patterns (e.g., *<person> VB:bear <date>*). To encourage wider coverage of patterns, they have used the canonical form of verbs, for example, the word *born* is normalized to *bear*. Varma et al. (2010) applied the POS tags, named entities, sentence window length and the order of occurrence of entities for constructing extraction rules. Xu et al. (2011) manually generated a list of trigger words and identified relations between entities if sentences contain the trigger words.

The benefit of these approaches is that no annotated data is needed, but the creation of reliable extraction patterns requires a great deal of expertise and a pattern-based method has a generalization issue over heterogeneous data.

2.3.2 Machine Learning Methods for PPE

Supervised machine learning methods can be used for PPE if manually labeled data is available. Kambhatla (2004) trained a log-linear classifier that incorporates lexical and syntactic features, including the words between entities, path between entities in a parse tree. Culotta et al. (2006) treated the relation extraction as a sequential labeling problem. Their supervised method uses features of syntactic information and relation patterns to improve system performance. To explore the rich representation of data, kernels are used to define similarity between entities in a high dimensional space. Tree kernels and shortest path dependency kernels are designed to calculate similarity based on shallow parse tree of text and have been used with SVMs and Voted Perceptron to extract relations (Zelenko et al., 2003; Bunescu and Mooney, 2005a, 2006). The difficulty of these approaches lies in the fact that manually annotated data are expensive to obtain, and most approaches rely on syntactic features (dependency parsing tree of a sentence).

Recently, a distant supervision approach has greatly attracted researchers' interest, since supervised relation extractors can be learned from a large number of facts in an existing knowledge base with few annotation efforts (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2010, 2011; Hoffmann et al., 2011; Sun et al., 2011). Freebase, a community-curated database that contains facts about named entities and their relations, is often used as the knowledge base to automatically generate labeled relation extraction training data. Researchers find entities and their relations from Freebase and then extract sentences containing these entities from Wikipedia articles. These sentences from Wikipedia with corresponding relation labels will be used as training data for relation extraction. Mintz et al. (2009) trained a logistic classifier using lexical features, lexical order, number of words, window size and syntactic features. Using the same set of features as Mintz et al. did, Riedel et al. trained a graphical model that explicitly decided whether two entities are related on the basis of *at least one sentence* assumption (Riedel et al., 2010). Since Mintz et al.'s model and Riedel et al.'s model assumed no overlapping relations, Hoffmann et al. allowed overlapping relations between entities, which has seen its superiority in experiments (Hoffmann et al., 2011), but performance still suffers due to the noise in training data when matching the content of an existing knowledge base to the corresponding Wikipedia articles. Similarly, Surdeanu et al. (2010) applied simple distance-based supervision for extracting personal profiles. They trained a multi-class logistic regression with L2 regularization classifier with such features as lexical feature, textual order, number of words, window size, syntactic features and so on. Later, Surdeanu et al. (2011) added new features, including an inference filter that discards attributes that do not support world knowledge constraints, a system combination model that votes between 10 different systems trained on different fragments of the knowledge base and incorporation of snippets and co-reference chains into training. Sun et al. (2011) continued to use the distant supervision approach for extracting personal profiles. However, they added the function of class label refinement to solve the problem of false negative examples in the training corpus.

Another approach related to our work is the joint modeling of entity and relation extraction. Roth and Yih used separate classifiers to find possible entities and relations, and then computed a most likely globally consistent global set of entities and relations using linear programming (Roth and Yih, 2007). Kate and Mooney (2010) took a joint approach to extract entities and relations from a sentence using a card-pyramid parsing technique which treats all candidate entities as the leaves and then builds a tree that encodes a relation between a pair of entities and a *NONE* relation if there is no relation between the two entities. They then used Support Vector Machines (SVMs) to predict the entity types and relations jointly.

2.3.3 Summary for PPE

This chapter gives a brief literature review of personal profiling/relation extraction. Generally, a simple rule-based method and machine learning approach are taken to extract relational facts from web texts in PPE. A simple rule-based approach has its own advantage in that no annotated data is needed, but the creation of reliable extraction patterns requires a great deal of expertise. Additionally, manual compilation of trigger words is time-consuming and a pattern-based method has a generalization issue over heterogeneous data. Supervised methods boost performance in PPE, but rely heavily on annotated training data which are costly to yield and domain sensitive. This leads to the study of the distant supervision approaches which do not require manual efforts in data annotations. It uses the external resource (Freebase, in particular) to extract relational facts for learning relation extractors. This approach can reduce the manual efforts of data annotation, but selecting quality training data in the knowledge base becomes a very important issue because two entities coexisting in a sentence may not express a relation even if they do have one in reality.

2.4 Named Entity Linking

To facilitate knowledge population, disambiguated persons with profiles can be linked to named entities in an existing knowledge base (McNamee et al., 2010; Ji et al., 2011; Ellis et al., 2012). This process is called named entity linking (NEL in short). Recently, Named Entity Linking (NEL) has drawn a great deal of attention from researchers due to its wide applications in linking patient health records, preventing identity crimes and so on (Rao et al., 2011). In general, NEL involves two steps: candidate generating and candidate ranking.

2.4.1 Candidate Generating and Ranking

Candidate generation produces a list of candidate entities to which the name mentions can be linked. For this purpose, some systems used simple query expansion methods for candidate generation (Chen et al., 2010). Most systems used bold texts in the first paragraph, Wikipedia redirects and disambiguation pages, anchor texts and search engines to generate candidate entries (Fern et al., 2010; Lehmann et al., 2010; Radford et al., 2010; Varma et al., 2010).

From the candidate list, a ranking approach can be applied to select the most likely entity for the target name mention. To reach this goal, some systems have treated it as an information retrieval procedure. Varma et al (2010) used a TFIDF weighting scheme to rank the candidates. Fern et al. (2010) applied the PageRank approach to calculate the ranking scores of entities based on the concurrence information of other entities. Simultaneously, many other systems used a supervised learning method. Agirre et al. (2009) trained a multiclass classifier to distinguish possible Wikipedia articles for the target name mention. Varma et al. (2009) applied the Rainbow Text Classifier to map the query to the most possible candidate. Li et al. (2009) employed a Listwise "Learning to Rank" model and the Naïve Bayes model to rank candidate entities. McNamee et al. (2009) took the SVM-rank learning approach to select the best knowledge base node. Zhang et al. (2010, 2011) proposed a system of using Lucene-based ranking, SVM-rank and binary SVM classifier for entity linking. They used the SVM-rank approach to choose the best candidate entity and the binary SVM classifier to validate whether the highest ranked candidate is believed to be the knowledge base node for the target name mention.

2.4.2 Summary for NEL

NEL maps the disambiguated persons with profiles to named entities in an existing knowledge base (Wikipedia). However, entity ambiguity (e.g. the mention "*John Howard*" can refer to the prime minister or the martial artist) is rather challenging for NEL. To solve this problem, both ranking methods and classification approaches are proposed. Nevertheless, these approaches would face the problem that the target entity in Wikipedia usually has only one corresponding article, resulting in an imbalanced data for learning a classifier.

Chapter 3 Graphical Models

Both WPD and PPE in this thesis are built within the framework of graphical models, which can provide sequential structure for semantic feature extraction and tree structure for both name disambiguation and profile extraction. Extraction of semantic keywords for WPD uses the linear-chain Conditional Random Fields (CRFs) model in Chapter 4. Ambiguity of person names is resolved by means of a hierarchical co-reference graphical model in Chapter 5. Personal profile is extracted using a general graphical model which considers two semantic constraints including trigger words and entity types in Chapter 6. Therefore, prior to the detailed explanation of these three major works in this thesis, it is necessary to give a brief introduction to graphical models as well as their inference and parameter estimation.

3.1 Introduction to Graphical Models

In many natural language processing applications, graphical models provide a natural way of exploring the interactions between hidden and observable variables. For example, in a named entity recognition task, the hidden variables can be the labels of entity types, such as Person, Organization, Location or Others; and the observation variables can be words in a sentence, prefixes, suffixes or word initials. Without loss of generality, let $\boldsymbol{x} = x_1, x_2, \dots, x_t, \dots, x_T,\ 1 \le t \le T$ be the sequence of input sentence with length being T, and $\boldsymbol{y}$ is a corresponding sequence of output labels. This representation can be plotted in Figure 3.1.

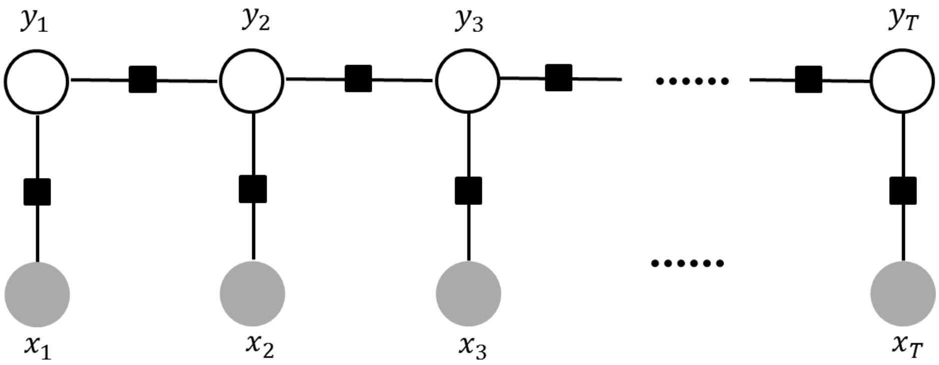


Figure 3.1 Graphical Representation between Hidden and Observable Nodes in a Sequential Chain Structure.

In Figure 3.1, the dark circles are observable variable nodes and the blank circles are hidden variable nodes. The shaded black boxes are factor nodes that link the observable and hidden nodes in the factor graph[2]. A factor ψ is actually a function that measures the compatibility between inputs and outputs. It is often defined as,

$$\psi(\boldsymbol{x},\boldsymbol{y}) = exp\left\{\sum_{k} \theta_k f_k(\boldsymbol{x},\boldsymbol{y})\right\}$$

where θ is the parameter for the k^{th} feature f_k. A feature can take the following form,

$$f_k(\boldsymbol{x},\boldsymbol{y}) = \begin{cases} 1, if\ y_i = PER\ and\ x_i = Clinton\ and\ x_{i-1} = Bill \\ 0, otherwise \end{cases}$$

This example feature returns 1 if y_i is *PER* and there are two observable words: *“Bill”* and *“Clinton”*; and 0 otherwise. By definition, the conditional distribution of $\boldsymbol{y}$ given $\boldsymbol{x}$ can be formulated as,

[2] A factor graph is a bipartite graph in which a variable node is connected to factor node if it is an argument to the factor.

$$p(\boldsymbol{y}|\boldsymbol{x}) = \frac{1}{Z(\boldsymbol{x})}\left\{\prod_{a} \psi_a(\boldsymbol{x}_a, \boldsymbol{y}_a)\right\}$$

where a denotes the a^{th} factor and $Z(\boldsymbol{x}) = \sum_{\boldsymbol{x},\boldsymbol{y}} \prod_a \psi_a(\boldsymbol{x}_a, \boldsymbol{y}_a)$ is a partition function that sums up all possible output sequences for the input $\boldsymbol{x}$. It can be seen that graphical models represent a distribution over a large number of variables by a product of factors (named as local functions) that each depends on only a small number of variables locally. This sequential structure of the graphical model has been widely used for natural language applications, for example, named entity recognition, part-of-speech tagging, Chinese word segmentation and so on (Lafferty et al., 2001; McCallum and Li, 2003; Peng and McCallum, 2004; Tseng et al., 2005; Sutton and McCallum, 2006; Zhao et al., 2011).

Graphical models have great flexibility in representation. We can extend the previous sequential structure into a tree structure as plotted in Figure 3.2.

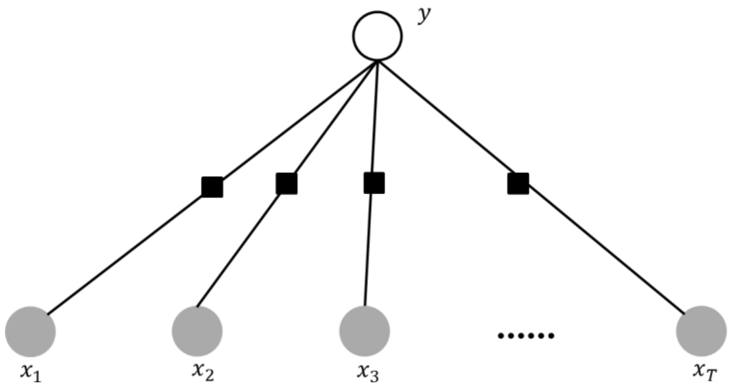


Figure 3.2 Graphical Representation in a Tree Structure

In Figure 3.2, the output is a single class label instead of a sequence of labels. This can be applied to classifying documents which are represented by observable nodes (actually words in documents). It can also be applied to relation extraction in which case the label y refers to a relation type (*birthdate*, *spouse*, *parents*, etc.) whereas the $x_1, \dots, x_T$ can be a sequence of lexical words that describe the relation, for example, *born*, *wife*, *father*/*mother* or *son/daughter*.

3.2 Inference and Parameter Estimation

By the conditional probability distribution $p(\boldsymbol{y}|\boldsymbol{x})$ defined in section 3.1, we can make predictions for $\boldsymbol{y}$ given a new input $\boldsymbol{x}$. This procedure is also called inference. Essentially, inference is to find the most possible label assignment for the input $\boldsymbol{x}$, which can be formally described as $\widehat{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}} p(\boldsymbol{y}|\boldsymbol{x})$. To solve the inference problem exactly, dynamic programming techniques are used. Rabiner (1989) introduced forward-backward and Viterbi algorithms for Hidden Markov Models (HMMs), which can be applied to linear chain CRFs (Lafferty et al., 2001). Belief propagation algorithm passes messages between factor and variable nodes where each message encodes information about the most likely assignment to subsets of variables (Yedidia et al. 2004). However, exact inference approaches can be computationally expensive and intractable for large and complex graphical models. For this reason, approximate inference approaches are studied, such as Markov Chain Monte Carlo (MCMC) sampling algorithm and variational methods. MCMC algorithm samples from a Markov chain that in the limit produces samples from the true distribution $P(\boldsymbol{Y}|\boldsymbol{X})$ (Robert and Casella, 2004). Variational methods attempt to find a simpler distribution that closely matches the target distribution (Jordan et al., 1999).

Parameter estimation involves finding a set of parameters $\Lambda = \{\theta\}$ that optimize the model, that is to say, the training data has the highest probability under this optimal model. Maximum likelihood method learns parameters by which the training data can have the highest probability. It is intractable in general because the partition function in the model is a sum over all possible label assignments. Therefore, approximate methods such as conjugate gradient or LBFGS are proposed (Nocedal and Wright, 1999; Wallach et al., 2002; Sha and Pereira, 2002). However, computing gradients in these methods requires inferences over the full dataset before parameters are updated, which can be computationally expensive. Instead of scanning all of the training data, parameters can be updated after seeing a few examples like the stochastic gradient descent (SGD) (Vishwanathan et al., 2006). In SGD, a training example is picked randomly at each iteration and then gradient of parameters will be computed for this example. An alternative approach is online learning which updates parameters after finding the best assignment under the current parameter configuration. The perceptron and MRIA updates are of this kind (Collins, 2002; Crammer et al., 2006). However, finding the best assignment can be computationally intensive in real applications (Culotta, 2008). SampleRank circumvents this problem by updating parameters when the model ranking of a pair of samples disagrees with the objective ranking (Culotta, 2008; Wick et al., 2009). It updates parameters after each sample and can quickly find a good set of parameters.

Different from the online learning which computes gradients of parameters against the ground truth, SampleRank applies some objective function to compute gradients between consecutive samples by performing parameter updates within each step of the MCMC inference (Wick et al., 2009). It computes gradients between neighboring configurations in an MCMC chain. Suppose that at the current time step, we have sample $\boldsymbol{y}$ and let $\boldsymbol{y}'$ be the sample at the successive time step in the chain. Let the model ranking of $\boldsymbol{y}$ and $\boldsymbol{y}'$ be $p_{\Lambda}(\boldsymbol{y}|\boldsymbol{x})$ and $p_{\Lambda}(\boldsymbol{y}'|\boldsymbol{x})$. They are un-normalized conditional probabilities. The objective ranking uses a loss function to measure a distance between the ground-truth assignment $\boldsymbol{y}^*$ and the pair of samples $\boldsymbol{y}$ and $\boldsymbol{y}'$, that is, $L(\boldsymbol{y}, \mathbf{y}^*)$ and $L(\boldsymbol{y}', \boldsymbol{y}^*)$. Disagreement occurs between model ranking and objective ranking when a higher model score is assigned to the sample that has a lower loss,

$$p_\Lambda(\boldsymbol{y}'|\boldsymbol{x}) > p_\Lambda(\boldsymbol{y}|\boldsymbol{x}) \wedge L(\boldsymbol{y}', \boldsymbol{y}^*) \ > L(\boldsymbol{y}, \boldsymbol{y}^*) \qquad \text{or}$$

$$p_\Lambda(\boldsymbol{y}'|\boldsymbol{x}) < p_\Lambda(\boldsymbol{y}|\boldsymbol{x}) \wedge L(\boldsymbol{y}', \boldsymbol{y}^*) \ < L(\boldsymbol{y}, \boldsymbol{y}^*)$$

In the above two cases, parameters will be updated as follows:

$$\Lambda^{\mathrm{t}+1} = \ \Lambda^{\mathrm{t}} + \eta(\emptyset(\boldsymbol{y}', \boldsymbol{x}) - \emptyset(\boldsymbol{y}, \boldsymbol{x}))$$

where $\emptyset$ refers to a collection of feature functions between output labels and sequence of inputs. η is the learning rate which will be adjusted by the MIRA approach (Crammer et al., 2006) or AdaGrad (Duchi et al., 2011).

3.3 Chapter Summary

In this chapter, we give a quick review of graphical models, inference algorithms and parameter estimation. It can be seen that graphical models provide a natural way to model the hidden and observable variables and can predict hidden variables or learn parameters using approximate methods in a large and complex graph. It can estimate parameters using ranking-based methods, SampleRank specifically. For these reasons, we have explored the sequential structure in extracting keywords as WPD features, since keywords can contain more than one tokens and the CRFs model can well capture the sequential structure of keywords. In WPD, we investigate the hierarchical structure between name mentions using the co-reference resolution technique which recursively partitions entities into a tree structure with latent sub-entities as child nodes and person names as observable leaf nodes. For PPE, a general graphical structure is built by incorporating two semantic constraints of trigger words and entity types in personal profiling extraction.

Chapter 4 Feature Extraction for WPD

To conduct WPD, meaningful features are important for separating persons having the same name. Previous works use a combination of different features such as tokens in text, URLs or titles in HTML documents, *n*-grams, snippets and so on (Bagga and Baldwin, 1998; Chen et al., 2009; Long and Shi, 2010). These features can introduce noise in WPD and most of them are extracted using unsupervised methods. Few researchers investigate supervised approaches to extract WPD features, because it is costly to obtain manually annotated training data and most existing corpora are confined to a specific domain and small in size. To solve this problem, we attempt to use naturally annotated corpus, on which we train supervised learning model to extract semantically more relevant information units as WPD features.

4.1 Using Keyword as Features

We consider keywords, which can be a single word or compound words (as key phrases) are semantically more meaningful WPD features. That is using “World Cup” is more meaningful than using “World” and “Cup” separately. For practical reasons, we cannot simply use a list of predefined keywords as WPD features. For this reason, Wikipedia anchor texts are good resource for training purposes, because they are created by crowds of contributors manually, thus are more semantically meaningful and reliable. Another advantage of using Wikipedia anchor texts is that it is less domain sensitive due to wide coverage of Wikipedia.

Instead of using simple statistics methods as reviewed in the Section 2.1.1 which can be restricted by low frequency terms, we choose the graph-based supervised machine learning method based on the Conditional Random Fields (CRFs) model. Compare to other machine learning algorithms such as Naive Bayes or decision tree classifier (Frank et al., 1999; Turney, 2000; Lopez and Romary, 2010; Nguyen and Luong, 2010), the CRFs model can treat keywords as a sequence to maintain their semantic integrity. It can also obtain keywords contained in the relevant sentences directly without the need to go through the entire document which may introduce more noise.

4.2 Algorithm Design for KWE

In this section, we first explain our method to generate keyword extraction training data automatically using a naturally annotated resource. We will then explain the CRFs model and feature design for keyword extraction and the post-ranking techniques for selecting final keywords.

4.2.1 Automatically Generating Training Data

Keyword extraction can take supervised training approaches. If training data is available, then statistical learning methods can be developed. However, manual annotation of training data is time-consuming and labor-intensive. To solve this problem, anchor texts in Wikipedia articles are used to automatically produce training data for keyword extraction. Anchor texts are used because they are created by crowds of contributors, and are thus both meaningful and reliable. Take the following sentence in the Wikipedia article (CERN) as an example,

More recently, CERN has become a centre for the development of ***grid computing****, hosting projects including the* ***Enabling Grids for E-sciencE*** *(EGEE) and* ***LHC Computing Grid****.*

The three anchor texts highlighted in bold: "***grid computing***", "***Enabling Grids for E-sciencE***" and "***LHC Computing Grid***" can be extracted as keywords. These keywords and the sentences in which they are located can be used to produce a keyword extraction model. The issue, however, is which anchor text can be used for keyword extraction since there are millions of anchor texts in Wikipedia. In this thesis, we hypothesize that the keywords in the testing documents would appear either partially or completely in the training text. Based on this assumption, *n*-grams are created from the testing documents and are checked against the anchor texts in Wikipedia articles. For example, given the below testing sentences

Efficient discovery of grid services is essential for the success of grid computing.

"*efficient discovery*", "*grid services*", "*grid computing*", "*success*", "*grid*", "*services*", and "*computing*" can be extracted first as keyword candidates. We then search for training sentences in the Wikipedia articles where they appear as anchor texts. All anchor texts in these sentences are then used to train the CRFs model. As examples, the following three training sentences are extracted based on this principle,

Knowledge discovery "On the Grid" generally refers to conducting knowledge discovery in an open environment using ***grid computing*** *concepts, allowing users ...*

Open Grid Services Architecture (OGSA) describes a ***service-oriented architecture*** *for a* ***grid computing*** *environment for business and scientific use.*

In ***computing****, the Oracle Application Server 10g (the "g" stands for* ***grid****) (short Oracle AS), consists of an integrated, standards-based* ***software platform****.*

Prior to the model training, we analyze the structure of a test document to select informative sentences from different sections. According to Edmundson (1969), "*the* *title circumscribes the subject matter of a document, ..., and topic sentences tend to occur very early or very late in a document and its paragraphs*". Based on this assumption, we select training sentences from structural sources: title, abstract, introduction, related works, conclusions and the first two sentences of paragraphs in the main body of text. Then, *n*-grams will be generated by the following steps:

- Extract all *n*-grams up to seven words;
- Remove *n*-grams starting/ending with a stop word, punctuations or numbers;
- Unigrams are constrained to be nouns;
- The first token and last token should be either a noun or an adjective in those *n*-grams ($n \geq 2$);
- Normalize *n*-grams by lowercasing and lemmatizing.

These *n*-grams will be checked against the anchor texts in Wikipedia to automatically create keyword training data.

4.2.2 KWE using the CRFs Model

To make use of these sentence-based training data, the CRFs model will be applied for extracting keywords from testing documents. The advantage of the CRF model is that it does not need to use the full document and a document set to collect frequency or document frequency features which are used by Naive Bayesian model, SVMs (Frank et al., 1999; Zhang et al., 2008; Xu et al., 2012). The CRFs model in this thesis depends on the contextual features of the candidate keywords including previous/next word adjacent to the current word and orthographic features such as prefix/suffix of a word, etc. Besides, Zhang et al. (2008), who applied the CRFs model for Chinese keyword extraction, reported that the CRFs model outperforms the SVMs in their experiments because its sequential labeling approach can better maintain the semantic integrity of the keyword sequence.

To use the CRF model, let $\boldsymbol{x} = x_1, x_2, \dots, x_t, \dots, x_T,$ where $1 \leq t \leq T$ and $\boldsymbol{x}$ is an input sequence with length being T and let $\boldsymbol{y}$ be a corresponding sequence of output labels. Each observation x_t is associated with a label $y_t \in${*B-KW,I-KW,O*} which indicates whether x_t is part of a keyword. *B-KW* indicates the beginning of a keyword, *I-KW* is the continuation of a keyword and *O* means the word is not part of the keyword. Then, the CRF model to label sequential data is defined by,

$$p(\boldsymbol{y}|\boldsymbol{x}, \Lambda) = \frac{\prod_{t=1}^{T} exp\{\sum_{k=1}^{K} \theta_k f_k(y_t, y_{t-1}, \boldsymbol{x}_t)\}}{\sum_{y'} \prod_{t=1}^{T} exp\{\sum_{k=1}^{K} \theta_k f_k(y'_t, y'_{t-1}, \boldsymbol{x}_t)\}}$$

where θ_k is the weight for the k^{th} feature function $f_k(y_t, y_{t-1}, \boldsymbol{x}_t)$. The parameters $\Lambda = \{\theta_k\}$ will be estimated to yield best sequence labels $\hat{\boldsymbol{y}}$ among all possible sequences for the input sequence $\boldsymbol{x}$. However, learning parameters in complex factor graphs is challenging because computing gradients requires inferences over the full dataset before the parameters are updated. The SampleRank method remedies this problem by performing parameter updates within each step of the MCMC inference (Wick et al., 2009). It computes gradients between neighboring configurations in an MCMC chain. Parameters are updated when the model ranking of any pair of samples disagrees with the ranking by the objective function. Suppose at the current time step *t*, we have the sample $\boldsymbol{y}_t$ and let the previous sample be $\boldsymbol{y}_{t-1}$ at time step *t-1* in the chain, model parameters are updated in case of disagreement of objective function and model ranking,

$$\Lambda^{\mathrm{t}} = \Lambda^{\mathrm{t}-1} + \eta(\emptyset(\boldsymbol{y}_t, \boldsymbol{x}) - \emptyset(\boldsymbol{y}_{t-1}, \boldsymbol{x}))$$

where $\emptyset: Y \times X \to \mathbb{R}^{|\Lambda|}$ refers to feature functions between labels and sequence of inputs. η is learning rate which will be adjusted by the AdaGrad approach with the Hamming loss function (Duchi et al., 2011).

Features for the CRFs model, based on both the surface forms of words and their part-of-speech tags (POS) include,

(1) w_0: current word

(2) w_{-2}: second previous word

(3) w_{-1}: previous word

(4) w_{+1}: next word

(5) w_{+2}: second next word

(6) $w_{-1}w_0$: previous word and current word

(7) w_0w_{+1}: current word and next word

(8) $w_{-1}w_{+1}$: previous word and next word

(9) t_0: POS tag of current word

(10) t_{-2}: POS tag of second previous word

(11) t_{-1}: POS tag of previous word

(12) t_{+1}: POS tag of next word

(13) t_{+2}: POS tag of second next word

(14) $t_{-1}t_0$: POS tags of previous, current words

(15) t_0t_{+1}: POS tags of current and next words

(16) $t_{-1}t_{+1}$: POS tags of previous, next words

(17) $t_{-2}t_{-1}t_0$: POS tags of previous two words and current word

(18) $t_0t_{+1}t_{+2}$: POS tags of current word and next two words

(19) $t_{-1}t_0t_{+1}$: POS tags of previous, current and next words

(20) $t_{-1}w_0$: current word and POS tag of previous word

(21) $w_0 t_{+1}$: current word and POS tag of next word

4.2.3 Post-ranking Keywords

In this post-ranking stage, we use the TFIDF ranking with an additional distance-based TFDIST method to select informative keywords in a document. Before post-ranking keywords generated by the CRFs model, we concatenate shorter keyword sequences into longer ones based on the following rules:

- if a keyword is preceded by adjective/noun and followed by a noun, concatenate the preceding and the successive words to make a new keyword. For example, if "*Nonlinear*" and "*algorithm*" serve as the predecessor and successor of the keyword "*extrapolation*", "*Nonlinear extrapolation algorithm*" will be added as a new keyword candidate.
- if a keyword is either preceded by adjective/noun or followed by a noun, create a new keyword by concatenating the preceding or successive word.
- if two keywords occur together, create a new keyword by concatenating the two of them.

To post-rank keyword candidates, we use a two-step ranking scheme. In the first step, only TFIDF is considered. The TFIDF score of a keyword *W* is defined by,

$$\mathrm{TFIDF_W} = \log_{10}(1 + freq_W) \times \log_{10}(\frac{N}{n_W})$$

where $freq_W$ is the number of times that the keyword *W* occurs. *N* is the total number of documents in a given corpus (Wikipedia in this thesis) and n_W refers to the number of documents containing the keyword *W*. The top-ranked keyword candidates above a threshold K_{TFIDF} will be selected as candidates for further consideration.

Prior to the second step ranking $\mathrm{TFDIST_W}$ based on both the distance value and term frequency, we found the problem of double counting for shorter keywords since they can be contained in longer keywords. For example, "*web service*" appears as part of the keyword "*web service discovery*". To eliminate double counting, El-Beltagy and Rafea (2010) have proposed a method to reduce the count of a keyword if it is part of another keyword. According to El-Beltagy and Rafea (2010), the method is more effective to tackle the double counting problem only for certain top-ranked keywords rather than the entire list. For this reason, we introduce a threshold K_{DC} (*DC* refers to double counting) such that only the top K_{DC} candidates will be adjusted for double counting of frequency and $K_{DC} < K_{TFIDF}$. Candidates with adjusted frequency will then proceed to the second step ranking $\mathrm{TFDIST_W}$, which is defined by,

$$\mathrm{TFDIST_W} = |W| \times \frac{1}{log(1+P)} \times \log(1 + freq_W)$$

where $|W|$ refers to the number of tokens of W, P refers to the position of first occurrence of W within a document. In the TFDIST ranking, the position score descends as the keyword makes its first appearance closer to the end of a document since informative keywords tend to show up at the beginning of a document. We add the length of W since longer keywords favor much more specific meaning than shorter ones and should be given higher weights. The top-ranked keyword candidates above a threshold K_{TFDIST} will be selected as the final keywords.

4.3 Performance Evaluation

Experiments are conducted on three datasets[3]: (1) SemEval-2010 Task 5 dataset from full scientific articles, (2) Wiki-20 data from computer science articles, and (3) 500 abstract dataset provided by Hulth from the *Inspec* database. To evaluate keyword extraction performance, we follow the exact match evaluation metric, that is, the keywords in the answer set are matched with those produced by our system. Based on the exact matching, micro-averaged precision (**P**), recall (**R**) and F-score (**F**) are calculated. They are defined as,

$$Precision\ (\boldsymbol{P}) = \frac{number\ of\ correctly\ extracted\ keywords}{number\ of\ all\ extracted\ keywords}$$

$$Recall\ (\boldsymbol{R}) = \frac{number\ of\ correctly\ extracted\ keywords}{number\ of\ keywords\ in\ golden\ answer}$$

$$F_{-score}(\boldsymbol{F}) = \frac{2PR}{P+R}$$

When running the CRFs model over the three datasets, we rank the test examples by sampling 20 iterations with a low temperature of 0.0001 in the Gibbs sampler using the Factorie tool (McCallum et al., 2009). In this work, Wikipedia dump[4] with the timestamp: April 03, 2013 is used as the raw training corpus which has 4,064,234 articles. The Stanford CoreNLP tool[5] is used to preprocess the articles including tokenization, part-of-speech tagging, and named entity recognition. To speed up the searching process, we build up indices for anchor text and whole texts of the Wikipedia articles using Lucene[6].

[3] https://github.com/snkim/AutomaticKeyphraseExtraction.

[4] http://dumps.wikimedia.org/enwiki/20130403/enwiki-20130403-pages-articles.xml.bz2

[5] http://nlp.stanford.edu/software/corenlp.shtml

[6] http://lucene.apache.org/

We then use the corresponding PDF files of the SemEval-2010 Task 5 dataset and Wiki-20 dataset to extract the structural information including titles, abstracts, introductions, related works, main body text and conclusions. In order to find the maximum length for the *n*-grams, we first investigated the distributions of *n*-grams, in three golden answers as shown in Table 4.1 and illustrated in Figure 4.1.

n-gram/ Dataset	**Total**	**n=1** (%)	**n=2** (%)	**n=3** (%)	**n=4** (%)	**n=5** (%)	**n=6** (%)	**n=7** (%)
SemEval-2010	1466	21.08	54.30	18.21	4.43	1.64	0.14	0.20
Wiki-20	321	45.79	45.79	7.48	0.93	-	-	-
Hulth2003	3847	14.74	54.07	23.24	5.77	1.51	0.57	0.11

Table 4.1 Total Number of *n*-grams and Percentage of *n*-grams in Three Golden Datasets

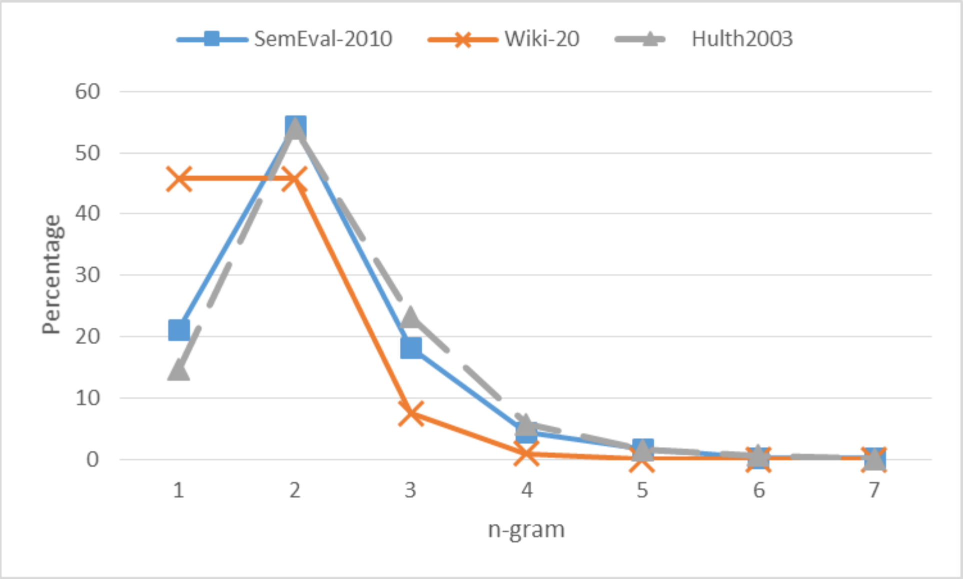


Figure 4.1 Distributions of *n*-grams in Three Golden Datasets

On the three datasets, bigram has the largest share and then the unigram and trigram have the second largest share. When *n* reaches 7, there are very few useful items. In fact, Wiki-20 has no items when *n* reaches 5. SemEval-2010 and Hulth2003 has only 0.2% and 0.11%, respectively. Therefore, for practical reasons, we choose 7 as the maximum length when extracting candidate *n*-grams.

For each *n*-gram, we select top 50 Wikipedia articles that contain this *n*-gram to identify the training sentences. For unigrams in a testing document, we use the top 50 unigrams based on their TFIDF scores. Table 4.2 lists the number of training documents available in the datasets and those documents in Wikipedia from which we extract the training sentences.

Dataset	**#Originally**	**#Transformed**
SemEval-2010	144	52,276
Wiki-20	20	18,988
Hulth2003	1,000	58,593

Table 4.2 Size of Training Data Transformed from Wikipedia

It is obvious that when using Wikipedia as a naturally annotated resource, we are able to obtain training data at a much larger size compared to manually annotated training data. The sentences that contain *n*-grams are used as the training data for the CRFs model. For reasons of convenience, we use the *KENAR* (initials of *Keyword Extraction using Naturally Annotated Resource*) to name our approach.

4.3.1 Experiments on the SemEval-2010 Task5 Data

In the SemEval-2010 dataset, the golden answer is based on 100 scientific articles provided with 15 keywords assigned by readers and authors. We first train the CRFs model for keyword extraction. Then, we use the two-step ranking schemes to obtain our answers. In the first step of the TFIDF ranking, we need to tune the threshold K_{TFIDF} to select top-ranked keywords. The result is given in Figure 4.2.

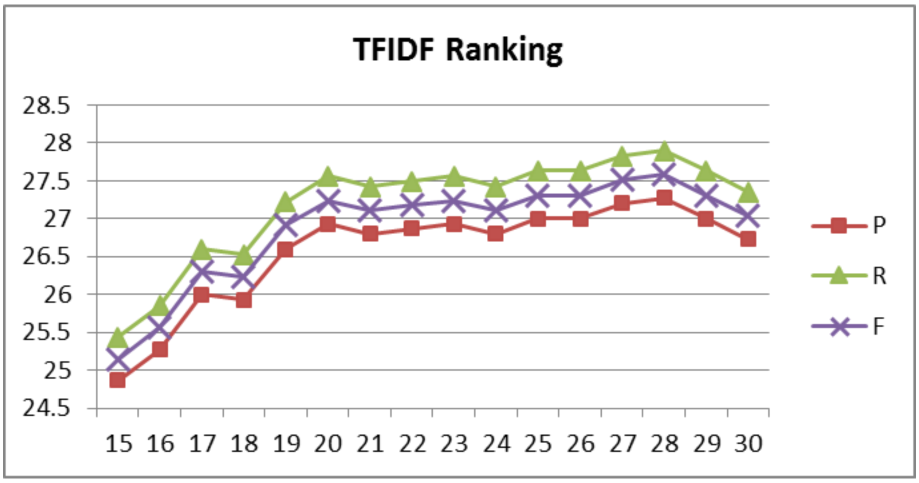


Figure 4.2 Effectiveness of Varying the K_{TFIDF} Values in TFIDF Ranking

It can be seen that the best performance in both precision and recall is when K_{TFIDF} =28. In the second step of the TFDIST ranking, we first eliminate the double counting of the shorter keywords that are contained in longer ones. Figure 4.3 shows the performance for different K_{DC} values.

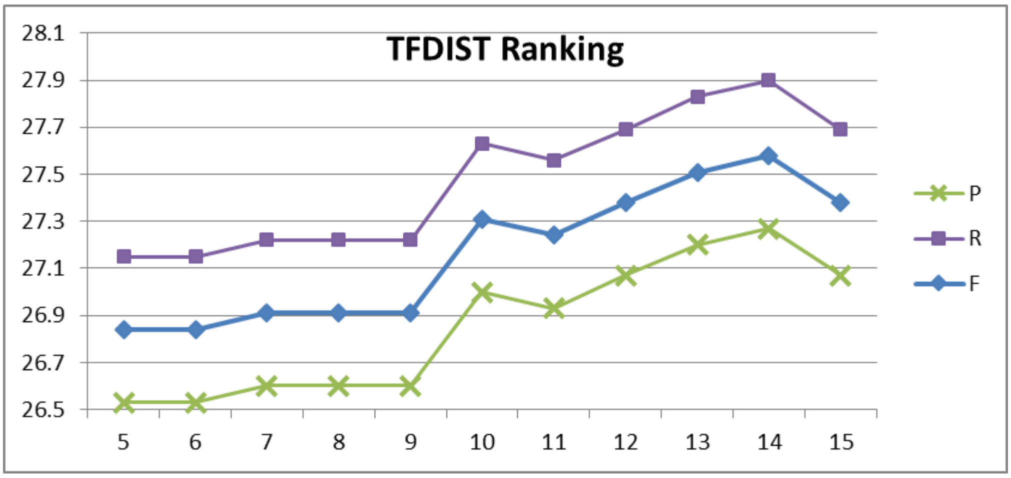


Figure 4.3 Effectiveness of Varying the K_{DC} Values in TFDIST Ranking

It can be seen that our algorithm is more effective when K_{DC}=14. Under this setting (K_{TFIDF}=28, K_{DC}=14), we selected top 15 keywords as the final answer (K_{TFDIST}=15). We then compared our results with the top three systems on the SemEval-2010 dataset by the top 5, 10 and 15 keywords.

Systems	**Top 5**		
	P (%)	**R** (%)	**F** (%)
HUMB	39.0	13.3	19.8
WINGNUS	40.2	13.7	20.5
KP-Miner	36.0	12.3	18.3
KENAR	39.0	13.30	19.8

Table 4.3 Comparison of Top 5 Keywords on SemEval-2010 Dataset

Systems	**Top 10**		
	P (%)	**R** (%)	**F** (%)
HUMB	32.0	21.8	26.0
WINGNUS	30.5	20.8	24.7
KP-Miner	28.6	19.5	23.2
KENAR	**32.10**	**21.9**	**26.04**

Table 4.4 Comparison of Top 10 Keywords on SemEval-2010 Dataset

Systems	**Top 15**		
	P (%)	**R** (%)	**F** (%)
HUMB	27.2	27.8	27.5
WINGNUS	24.9	25.5	25.2
KP-Miner	24.9	25.5	25.2
KENAR	**27.27**	**27.90**	**27.58**

Table 4.5 Comparison of Top 15 Keywords on SemEval-2010 Dataset

Table 4.3, Table 4.4 and Table 4.5 show that compared to the state-of-the-art *HUMB* system in F-scores, our *KENAR* system has gained a marginal increase in the top 5, 10, 15 keywords. The *HUMB* system uses structural, content, lexical/semantic features. When post-ranking candidate keywords, they have used the HAL repository of research publications with approximately 139,000 articles to select final keywords (Lopez and Romary, 2010). Additionally, they have used a large terminological database (GRISP) to extract lexical/semantic features. This implies that the *HUMB* system trains and post-ranks keywords using domain-specific data resources. Our system, on the other hand, is not confined to a specific domain in keyword extraction and post-ranking. The second best system *WINGNUS* extracts 19 features from structures of a document and then applies the Naïve Bayesian classifier to extract keywords (Nguyen and Luong, 2010). In comparison, our system gains higher F-scores on the top 10 and 15 keywords. Unlike their Naïve Bayes approach, our sequential tagging model in *KENAR* puts no limits on the length of a keyword. It can extract keywords of lengths from 1 to 5 on this SemEval-2010 dataset. The unsupervised *KP-Miner* system employed a set of rules to elicit candidate keywords, for example, frequency and position filtering. Then they introduced a boosting factor into keyword ranking to avoid the bias towards keywords of length 1. The benefit of this system is that it requires no training data, but it needs to tune the boosting factor (El-Beltagy and Rafea, 2010).

As mentioned previously, we have analyzed the structure of a document to select informative sentences from different sections: title (T), abstract (A), introduction (I), related works (R), conclusions (C), first two sentences of each paragraph in the body (B). Table 4.6 gives the F-scores of top 5, 10, 15 keywords by adding each section incrementally.

Systems	**F (%)**		
	Top5	**Top10**	**Top15**
T+A	18.92	24.17	23.33
T+A+I	19.84	26.04	**27.58**
T+A+I+R	20.34	26.19	27.44
T+A+I+R+C	**20.65**	**27.00**	**27.58**
T+A+I+R+C+B	20.24	25.22	25.96

Table 4.6 Performance using Different Structures from a Document

Table 4.6 shows that title, abstract, introduction, related work plus conclusion (T+A+I+R+C) give the best performance since they are more informative than the other sections in a long document. It is easy to see that the combination of title, abstract and introduction (T+A+I) has already reached the same F-score in top 15 keywords, although it has a marginally lower F-scores among the top 5 and 10 keywords. In this thesis, when making a comparison to other state-of-the-art systems, we simply select testing sentences from structural sources of title, abstract and introduction (T+A+I), because a good performance has already been guaranteed when using them in combination.

To test the effectiveness of the three rules for concatenating keyword sequences, we make a comparison between using and not using keyword concatenations under the setup (K_{TFIDF} =28, K_{DC} =14, K_{DIST} =15). Experimental comparisons of F-scores are given in Figure 4.4,

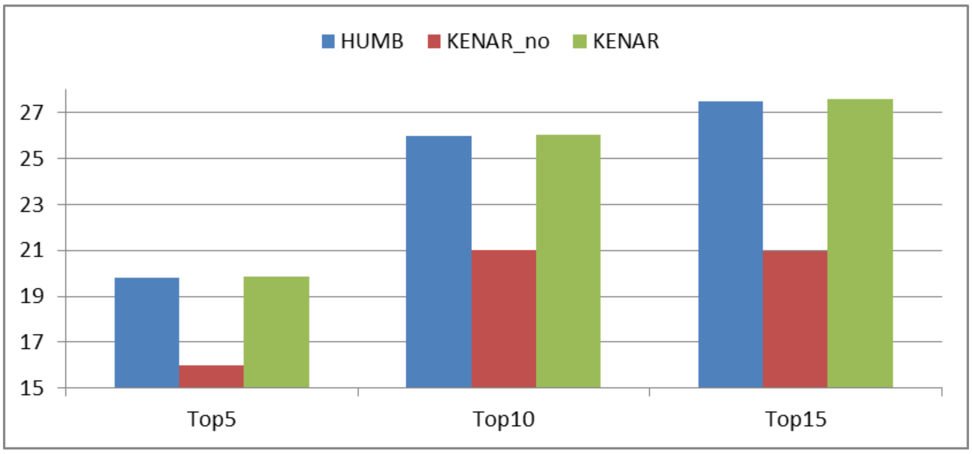


Figure 4.4 Comparison of Using and Not Using Keyword Concatenation on the SemEval-2010 Data

Figure 4.4 shows that using keyword concatenation rules can greatly boost F-scores. This means that some longer keywords, are recognized separately by multiple shorter keywords in the CRFs model. This is because some keywords cannot find training instances if they are combined. For example, we cannot find "*web service discovery*" from the Wikipedia anchor texts, but we can find "*web*" and "*service discovery*". Figure 4.4 also shows that $KNEAR_{no}$ is not doing as good as that of the *HUMB* system. The reason can be that our system is not trained on the manually annotated data. However, after incorporating the keyword concatenation, our system achieves a marginally better performance than the *HUMB* system, which is trained on the manually annotated data with a rich set of features.

For the two-step ranking, we also evaluated the effectiveness of each step separately. Experimental results of the F-scores are given in Table 4.7,

Systems	**F (%)**		
	Top5	**Top10**	**Top15**
HUMB	19.8	26.0	27.5
TFIDF Rank	18.82	24.17	25.15
TFDIST Rank	18.61	24.90	*27.04*
KENAR	**19.84**	**26.04**	**27.58**

Table 4.7 Comparison of Two Rankings on SemEval-2010 Dataset

In Table 4.7, the TFDIST ranking is better than the TFIDF ranking. This is desirable in the case of no external corpus to obtain document frequency for a keyword. When the two steps are combined, our system gives the best performance in F-scores for the top 5, 10 and 15 keywords. This is because the TFIDF step can select keywords that are informative for a document, and the TFDIST step will check which keywords come closer to the beginning of a document.

4.3.2 Experiments on the Wiki-20 Data

In this dataset, there are 20 documents, with each document being tagged by 15 teams of computer science students using Wikipedia article titles as the golden answer (Medelyan, 2009). All the answers provided by 15 teams are used in evaluation. In this experiment, we set K_{TFIDF} =35, K_{DC} =16 and K_{TFDIST} =18. For evaluation, we use the set of all keywords in the answer set, the same experimental setup used in Wu and Giles's work (Wu and Giles, 2013). Micro-averaged results are listed in Table 4.8,

Systems	**Wiki-20**		
	P (%)	**R** (%)	**F** (%)
KEA	18.4	21.5	19.8
CTI	19.6	22.7	21.0
KENAR$_{no}$	**26.89**	**28.08**	**27.47**
KENAR	21.78	23.66	22.73

Table 4.8 Comparison with CTI and KEA on Wiki-20 Data

In Table 4.8, *KENAR* shows a better performance when compared to the *KEA* and *CTI* systems. *KEA* builds a Naïve Bayes extractor using the TFIDF value, distance score, the length of a keyword and node degree (Frank et al., 1999). It uses 5-fold cross validation on the 20 documents and limits the length of a keyword to 3. *CTI* creates a context-aware term informativeness model using the semantic similarity/relatedness between the term's context and the term's featured context with highest authority scores in a knowledge base (Wu and Giles, 2013). Keywords are then ranked by the term informativeness. *CTI* limits the length of keywords to 4. In comparison, *KENAR* has no limits to the length of a keyword. It can extract keywords of lengths ranging from 1 to 5 in this Wiki-20 dataset. We also present a system $KNEAR_{no}$ that uses no concatenated keywords in Table 4.8. Note that $KNEAR_{no}$ has obtained the best performance. The reason can be that golden answers in the 20 documents are Wikipedia article titles which are contained in our automatically generated training data. Thus, without keyword concatenation, our CRFs model can well capture the keywords using the automatically generated training data. Otherwise, concatenation can only introduce more noise to decrease system performance.

Wu and Giles (2013) have also experimented with the SemEval-2010 dataset on top 15 keywords. The experimental results are given in Table 4.9,

Systems	**SemEval-2010**		
	P (%)	**R** (%)	**F** (%)
HUMB	27.2	27.8	27.5
CTI	19.3	20.1	19.7
CTI+	25.3	26.2	25.7
KENAR	**27.27**	**27.90**	**27.58**

Table 4.9 Comparison with CTI and HUMB on SemEval-2010 Data

Table 4.9 also includes the state-of-the-art *HUMB* system on the SemEval-2010 data. On this dataset, *CTI* does not perform as well as it does on the Wiki-20 data, achieving 19.7% in micro-averaged F-score. They then improve system performance by 6% (*CTI+*) adding the structural features: title, abstract, section titles and content, with weights manually set to 0.3, 0.4, 0.25 and 0.05, respectively. In comparison to *CTI+*, our *KENAR* system simply uses title, abstract and introduction, but obtains a higher F-score by 1.88%.

The effectiveness of the two ranking schemes is also checked on the Wiki-20 dataset, and results are given in Table 4.10.

Systems	**Wiki-20**		
	P (%)	**R** (%)	**F** (%)
KEA	18.4	21.5	19.8
CTI	19.6	22.7	21.0
TFIDF Rank	18.6	20.19	19.36
TFDIST Rank	20.18	21.77	20.94
KENAR	**21.87**	**23.66**	**22.73**

Table 4.10 Comparison of Two Rankings on Wiki-20 Dataset

Table 4.10 shows the two ranking schemes reach a closely equal performance and the *TFDIST Rank* has outperformed the *KEA* approach which uses the Naïve

Bayes classifier to extract keywords. Our *KENAR* system combines the two ranking approaches and obtains the best F-score, showing that TFIDF ranking can select a good set of initial keywords, while the TFDIST ranking can help determine which keywords are important for that document. Moreover, the TFDIST ranking is also better than the TFIDF ranking on the Wiki-20 dataset.

4.3.3 Experiments on Hulth's Abstract Data

This dataset contains 2,000 abstracts from the *Inspec* database of journal papers. The 2,000 abstracts are divided into 1,000 for training, 500 for validation and 500 for testing. In this thesis, we use the 500 testing abstracts. Each abstract has two sets of keywords: one set contains the controlled terms (terms restricted to the *Inspec* thesaurus) and the other set contains the uncontrolled terms which may not appear in the abstracts. Take a testing abstract as an example,

Fresh voices, big ideas [IBM internship *program]*

IBM is matching up computer-science and MBA students with its business managers in an 11-week summer internship program and challenging them to develop innovative technology ideas

The uncontrolled keywords are: “*internship program*”, “*IBM business managers*”, “*MBA college students*”, “*patents*”. From the above abstract, we can only find the consecutive keywords “*internship program*”. Thus, when calculating the recall, only the keywords present in abstracts are considered. This setup is the same as Hulth's experiment and Mihalcea and Tarau's TextRank model.

In this experiment, we set K_{TFIDF} =16, K_{DC} =11 and K_{TFDIST} =12. For comparison purposes, we directly use the keyword extraction results reported by Hulth (2003) and Mihalcea and Tarau (2004). The results are given in Table 4.11,

Systems	**Hulth's Abstract Data**		
	P (%)	**R** (%)	**F** (%)
Hulth_{2003}	25.9	51.7	33.9
TextRank (window=2)	31.2	43.1	36.2
TextRank (window=3)	28.2	38.6	32.6
KENAR$_{\text{no}}$	18.94	27.95	22.58
KENAR	**31.54**	**48.29**	**38.16**

Table 4.11 Comparison with Hulth_{2003} and TextRank on Abstract Data

In Table 4.11, *KENAR* obtains the highest precision and recall on this dataset. *TextRank* builds a graph of lexical units and an edge between two lexical units will be added into this graph if they co-occur within a specified window size. It achieves the best performance when the window size is 2. But when the window size is set to 3, the performance decreases by 3.6%. This is because the larger window size does not hold strong connection between words, but this also limits its capability in extracting longer keywords. Unlike the Hulth's model and *TextRank* model, *KENAR* can produce longer keywords using the CRFs sequential tagging model. From this dataset, *KENAR* can extract keywords of lengths ranging from 1 to 6. This ability is further reinforced by the keyword concatenation rules, an increase of 15.58% in F-score over $KNEAR_{no}$. One possible reason for the lower performance of $KNEAR_{no}$ could be that short abstracts are unable to collect more training data. Statistics in Table 4.2 shows that 500 abstracts have 58,593 training documents in total with each abstract having 117 training documents on average, but the 100 full-length SemEval-2010 documents have 52,276 documents with each document having 523 training documents on average.

It is also worth noting that different from the full texts in the SemEval-2010 and Wiki-20 datasets, Hulth's dataset contains only abstracts, with the number of tokens ranging from 23 to 338 (Hulth, 2003). Compared to the full-length documents, this number is small. However, keyword extraction in this case is more widely applicable, since many documents on the Internet nowadays are written in short texts.

4.4 Chapter Summary

To extract semantically relevant keyword features for WPD, in this thesis, we present a novel method for transforming Wikipedia anchor texts into keyword extraction training data. Anchor text is used due to its reliability and wide coverage of different domains. On the basis of this automatically generated training data, the CRFs model is applied to create the initial set of keyword candidates. Then, the two-step ranking (TFIDF and TFDIST) is used to select the most informative keywords for a document. In the evaluation of the SemEval-2010 dataset, we obtained a comparable performance to the state-of-the-art system. In the evaluation of the Wiki-20 corpus, we obtained a 22.73% average F-score, an increase of 1.73% compared to the state-of-the-art approach using term informativeness. Further evaluation on Hulth's short abstracts proves that our approach can obtain a 1.96% increase in F-score compared to the *TextRank* system. However, only word surface form and POS tags are used in the current CRFs model for keyword extraction. In future work, we plan to explore word frequency score and document structural features in the CRFs model.

A key contribution of our work is that it works well for both long and short documents. In addition, our approach is much less domain sensitive because the training data transformed from Wikipedia can have a large domain coverage. Our approach can also extract longer keywords using the sequential labeling model without strict limits on keyword length.

Chapter 5 Web Person Disambiguation

Based on the keyword features, Web Person Disambiguation (WPD) can be conducted to group a set of documents for a given name into different clusters, with each cluster referring to the same entity. The set of documents can be the search results of a search engine for a given name. In most cases, we assume that this set of documents are already locally stored and indexed at the server site of the search engine. This chapter will only focus on identifying the different named entities for a given set of documents which contain the mentions of the query name.

Since the contextual information for a person name is often sufficient to define an entity, we consider the contextual relevance of a feature in the Hierarchical Agglomerative Clustering (HAC) algorithm. However, the HAC algorithm needs to tune the threshold for selecting the number of clusters, which is a challenging problem in WPD since we do not know how many persons exist in the returned search results. To tackle this problem, we introduce a hierarchical co-reference resolution technique to disambiguate person names. Unlike the clustering methods, this disambiguation method does not need to tune the threshold for determining the number of persons, and can produce good quality clusters for each person. It simply resolves name ambiguities by deciding whether two mentions are co-referential or not in a factor graph through various feature representations.

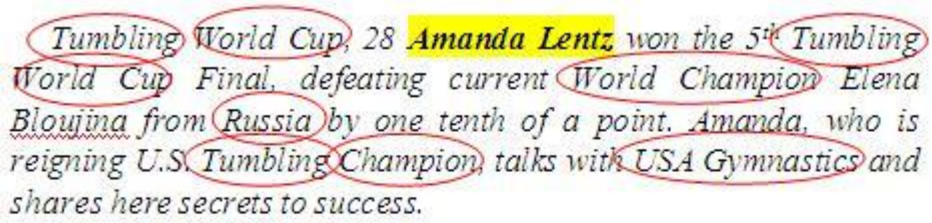
5.1 Contextual Relevance Weighting for WPD

5.1.1 Contextual Relevance Weighting

Rich features and external knowledge can improve the performance of WPD. However, mining rich and external information is rather expensive. Intuitively, as long as there is contextual information for a person name, it is often sufficient for a human to identify it. For example, the following is a short paragraph describing the person "*Amanda Lentz*",

Tumbling World Cup 28 ***Amanda Lentz*** *won the 5th Tumbling World Cup Final, defeating current World Champion Elena Bloujina from Russia by one tenth of a point. Amanda, who is the reining U.S. Tumbling Champion, talks with USA Gymnastics and shares here secrets to success.*

The words "*tumbling*", "*world*", "*cup*", "*won*", "*final*", "*defeating*", "*champion*" etc. surrounding the name mention can define this person "*Amanda Lentz*" as an athlete. In this thesis, we use these token features including noun, verb, and adjective. Although the token feature is limited to three types (noun, verb and adjective), it is still high dimensional. To resolve this problem, we explore the use of keywords for representing a person name. For example, the keywords surrounding the name mention *Amanda Lentz* in (Xu et al., 2012).



Tumbling World Cup, 28 ***Amanda Lentz*** *won the 5th Tumbling World Cup Final, defeating current World Champion Elena Bloujina from Russia by one tenth of a point. Amanda, who is reigning U.S. Tumbling Champion, talks with USA Gymnastics and shares here secrets to success.*

Figure 5.1 Keywords in the Context of "*Amanda Lentz*"

The keywords "*tumbling*", "*world cup*", "*world champion*" and "*usa* *gymnastics*” are more distinctive than tokens and can identify this person “*Amanda Lentz*” as an athlete as well. Meanwhile the number of keywords is less than that of tokens, thus realizing the goal of dimension reduction for a person name.

In succession, we attempt to make full use of contextual features that surround name mentions by assigning a higher weight to them. Our algorithm uses the standard Vector Space Model (VSM) for each document and the HAC algorithm is used to partition documents into different clusters.

5.1.2 HAC Clustering for WPD

HAC clustering algorithm treats each document as a singleton cluster at the start, and then merges pairs of clusters using different linkage metrics until all clusters are agglomerated into one cluster containing all documents. Commonly, it represents a document as a vector space model (VSM). In this section, the VSM simply uses the keywords extracted using the algorithm described in **Chapter 4**. Let V and V' be keyword vectors of two documents with the same entity name. The cosine similarity metric is defined as:

$$cosine(V, V') = \frac{V \circ V'}{|V||V'|}$$

Now, the question remains as how to weigh the keywords in each vector representation. In the VSM model, the TFIDF is the most commonly used weighting scheme, as it is a good indication of the importance of a term to a document in a corpus. To incorporate contextual relevance into the weighting scheme, we assume the terms within a context window of the name mention should be given higher weighted values. Let $\emptyset(t_i)$ denote the additional weight of i^{th} term t_i when contextual relevance is being considered, and let C_j be a set of contextual terms of a person name in j^{th} document. Then, the revised weighting scheme can be defined as,

$$W_{t_i,d_j} = \log\left(TF\left(k_{ij}\right) + \ 1\right) \times \log\left(IDF(k_i)\right) + \sum_{t_i \in C_j} \emptyset(t_i)$$

where

$$\emptyset(t_i) = \begin{cases} 1, if\ t_i \in C_j \\ 0, otherwise \end{cases}$$

$\sum \emptyset(t_i)$ sums up the number of times the term t_i occur around the name mention. In this sense, W_{t_i,d_j} will assign higher weight to a term if it is in the proximity where the person name is mentioned.

5.1.3 Performance Evaluation

The evaluation of the algorithm is conducted using the test data of WePS2 workshop 2009, which has 30 ambiguous names: 10 names sets from the 1990 U.S. census, 10 from participants in ACL'08 and 10 from Wikipedia. For each name, a web search is performed using Yahoo!API. The top results metadata are stored in an XML file for each name. This metadata includes the title, snippet and URL of each web result. Each name has approximately 150 documents. Golden answers for the test data are the manual clustering of the documents by human annotators.

To evaluate the performance of the algorithm, two sets of common performance measures for WPD are used: the BCubed precision and recall (BEP and BER in short), Purity and Inverse Purity (InvPurity in short) (Artiles et al. 2007, 2009). The purity score is given as follows.

$$Purity = \sum_i \frac{|C_i|}{n} maxPre(C_i, L_j)$$

$$Pre\left(C_i, L_j\right) = \frac{C_i \cap L_j}{C_i}$$

where C denotes the clusters produced by the system, L denotes the manually annotated categories and *n* the number of clustered documents. $Pre\left(C_i, L_j\right)$ refers to precision of a C_i for the category L_j. Inverse purity focuses on the cluster with the maximum recall for each category, defined by,

$$InvPurity = \sum_i \frac{|L_i|}{n} maxPre(C_i, L_j)$$

Finally, to rank the clustering performance, the harmonic F-score of purity and inverse purity is defined as follows:

$$F_\alpha = \frac{1}{\alpha \times \frac{1}{Purity} + (1-\alpha) \times \frac{1}{InvPurity}}$$

where α={0.2, 0.5}. If α=0.2, more importance will be given to the inverse purity, thus assigning a higher weight to recall. In the case of α=0.5, equal weighting will be given to precision and recall.

BCubed metrics calculate the precision and recall related to each item in the clustering result. The precision of one item represents the number of items in the same cluster that belong to its category, whereas the BCubed recall represents how many items from its category belong to its cluster. They are formally defined as,

$$Pre.BCubed = Avg_e\left[Avg_{e', C(e) \cap C(e') \neq 0}[Mult.Pre(e, e')]\right]$$

$$Recall.BCubed = Avg_e\left[Avg_{e', L(e) \cap L(e') \neq 0}[Mult.Recall(e, e')]\right]$$

$$Mult.Pre(e, e') = \frac{\min(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|C(e) \cap C(e')|}$$

$$Mult.Recall(e, e') = \frac{\min(|C(e) \cap C(e')|, |L(e) \cap L(e')|)}{|L(e) \cap L(e')|}$$

e and *e'* are two documents, *C(e)* and *L(e)* denote the clusters and categories related to *e*. The multiplicity precision $Mult.Pre(e, e')$ is 1 when *e* and *e'* in the same cluster share the same category. Therefore, the BCubed precision of one item is its averaged multiplicity precision with the other items in the same categories. The multiplicity recall $Mult.Recall(e, e')$ is 1 when *e* and *e'* in the same category share the same cluster. Similarly, the harmonic F-score of BCubed precision and recall is defined by,

$$F_\alpha = \frac{1}{\alpha \times \frac{1}{Pre.BCubed} + (1-\alpha) \times \frac{1}{Recall.BCubed}}$$

where α={0.2, 0.5}.

In this thesis, we tend to use keywords as semantically relevant features for WPD. Keywords are extracted from the test data of the WePS2 workshop 2009. We give a partial list of sample keywords extracted for person "*Amanda Lentz*" and "*Benjamin Snyder*" in Table 5.1.

Amanda Lentz	**Benjamin Snyder**
tv series	electrical engineering
sound clip	natural language processing
imdb	mit
team captain	phd student
writers	machine learning
midfielder	morphological analysis
ncaa college cup	information extraction
free kick	executive producer
athletic director	imdb database manager
high school band	photo gallery
penn state	james madison university
actors	delware superior court
dvds	attorney advertisement
movie trailer	case law
rowlings	horticulture

trampoline	wildlife
intercontinental judges	auto repair
olympic games	car rental
technical committee	customer service
world championship season	auto dealer
olympic champion	sale representative
……	……

Table 5.1 Sample Extracted Keywords

Table 5.1 shows that the extracted keywords are more informative and distinctive and are good candidates for WPD. For evaluation, we vary the window size of the contextual keywords and give the experimental results in Figure 5.2 and Figure 5.3.

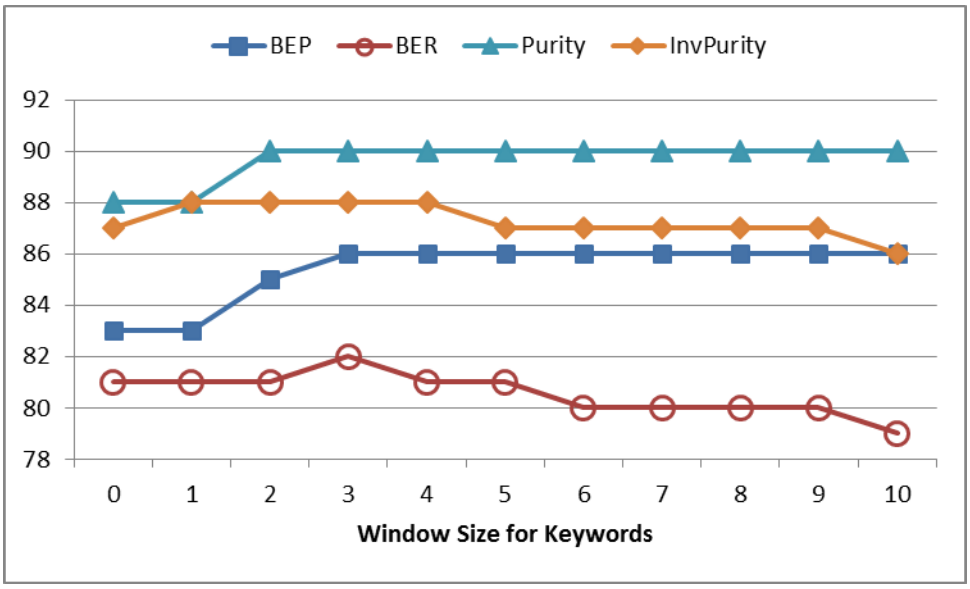


Figure 5.2 BCubed and Purity scores by Varying Window Size for Keywords

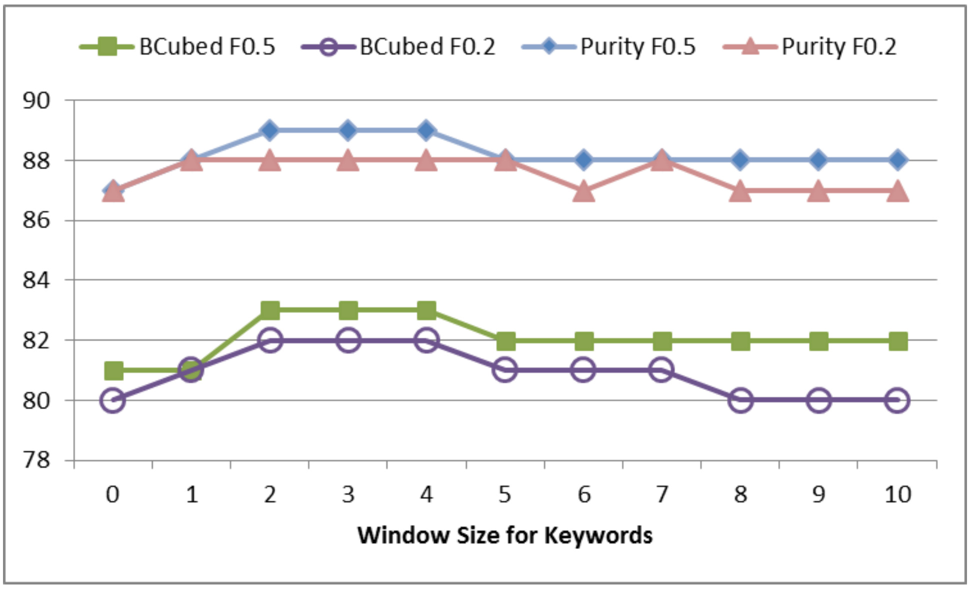


Figure 5.3 BCubed and Purity F-scores by Varying Window Size for Keywords

Figure 5.2 and Figure 5.3 show that when the number of contextual keywords is set to 3, both our system achieves the best BCubed and Purity scores. Hence, we label our system as $\boldsymbol{C_{keyword_3}}$. We also use $\boldsymbol{C_{keyword_0}}$ to represent no contextual relevance. To have a fair comparison, we directly use the threshold 0.1 which is tuned by the top performing system in the WePS2 workshop. We then compared our results with the top three systems in the WePS2 workshop. They are denoted by *T1: PolyUHK* (Chen et al. 2009), *T2: UVA_1* (Balog et al. 2009) and *T3: ITC_UT_1* (Ikeda et al. 2009). Table 5.2 and Table 5.3 show the performance of BCubed and Purity F-scores on the WePS2 dataset for the keywords based feature with respect to the use of contextual relevance.

Systems	**Macro-averaged Scores**			
	BCubed F-scores			
	α= 0.5 (%)	α= 0.2 (%)	BEP (%)	BER (%)
T1: PolyUHK	82	80	87	79
T2: UVA_1	81	80	85	80
T3: ITC_UT_1	81	76	93	73
$\boldsymbol{C_{keyword_0}}$	77	79	78	81
$\boldsymbol{C_{keyword_3}}$	**82**	**82**	85	**82**

Table 5.2 Comparison between Our Keyword-based Approach and Top Systems using BCubed Scores

	Purity F-scores			
	α= 0.5 (%)	α= 0.2 (%)	Purity (%)	InvPurity (%)
T1: PolyUHK	88	87	91	86
T2: UVA_1	87	87	89	87
T3: ITC_UT_1	87	83	**95**	81
$\boldsymbol{C_{keyword_0}}$	84	86	84	88
$\boldsymbol{C_{keyword_3}}$	**88**	**88**	89	**88**

Table 5.3 Comparison between Our Keyword-based Approach and Top Systems using Purity Scores

Table 5.2 and Table 5.3 show that $\boldsymbol{C_{keyword_3}}$ achieves a similar result with the *PolyUHK* system in BCubed and Purity F-scores when it uses contextual relevance weighting for the surrounding three keywords. When comparing to the features used in the top three systems, the *PolyUHK* system incorporates tokens, title tokens, n-gram features into its system. Besides, it has to tune the unigram and bigram weights through the Goodgle 1T corpus which contains English word n-grams and their frequency counts. The n-grams range from unigram to five-gram and their frequency counts are generated from approximately 1 trillion tokens of English web texts. Our system simply uses the TFIDF weighting scheme. The second best system *UVA_1* employs all tokens in the documents, and the third best system *ITC_UT_1* uses named entities, compound nouns and URL features. Our $\boldsymbol{C_{keyword_3}}$ system uses the keywords as features for WPD and add higher weights to keywords that are surrounding name mentions. In this case, it achieves the same results as the best system in BCubed and Purity F-scores. When compared to the *PolyUHK* system in terms of BCubed and Purity $F_{\alpha=0.2}$, we still gain one 2% and 1% increase, respectively.

In comparison to the $\boldsymbol{C_{keyword_0}}$ system, we found that $\boldsymbol{C_{keyword_3}}$ has obtained an increase of 5% in BCubed $F_{\alpha=0.5}$ and 3% in BCubed $F_{\alpha=0.2}$. $\boldsymbol{C_{keyword_3}}$ has also obtained an increase of 4% in Purity $F_{\alpha=0.5}$ and 2% in Purity $F_{\alpha=0.2}$. This demonstrates the effectiveness of our approach in capturing the information neighboring the person name mentions.

To verify the effectiveness of contextual relevance for token features, we run experiments using tokens of nouns, verbs and adjectives. We vary the window size of the contextual tokens and give the experimental results in Figure 5.4 and Figure 5.5.

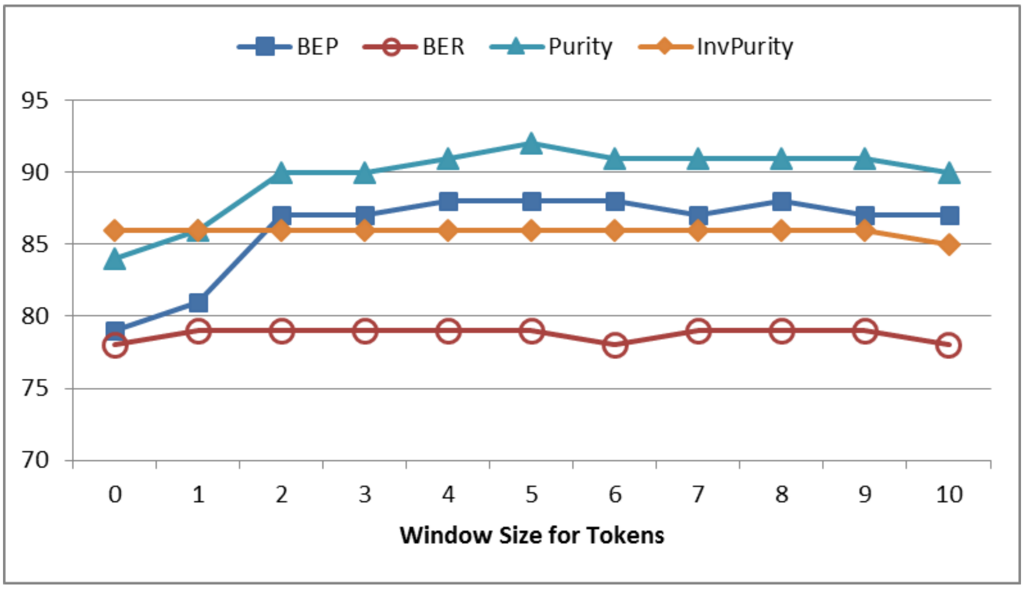


Figure 5.4 BCubed and Purity Scores by Varying the Window Size for Tokens

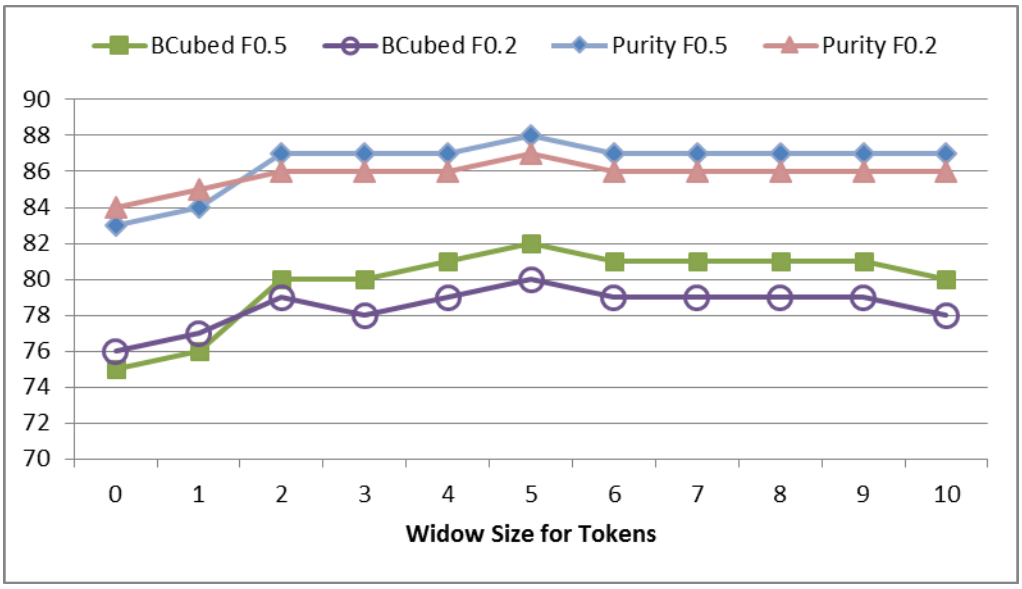


Figure 5.5 BCubed and Purity F-scores by Varying Window Size for Tokens

Figure 5.4 and Figure 5.5 show that when the number of contextual tokens is set to 5, our system obtains the best BCubed and Purity scores. In this set of experiments, we manually tune the threshold for the HAC algorithm to 0.135 when the highest BCubed and Purity scores are obtained. We give the experimental results with using and not using contextual relevance. The comparisons using BCubed and Purity scores are given in Table 5.4 and Table 5.5.

Systems	**Macro-averaged Scores**			
	BCubed F-scores			
	α= 0.5 (%)	α= 0.2 (%)	BEP (%)	BER (%)
T1: PolyUHK	82	80	87	79
$\boldsymbol{C_{token_0}}$	75	76	79	78
$\boldsymbol{C_{token_5}}$	81	80	87	79

Table 5.4 Comparison between Our Token-based Approach and Top System using BCubed Scores

Systems	**Macro-averaged Scores**			
	Purity F-scores			
	α= 0.5 (%)	α= 0.2 (%)	Purity (%)	InvPurity (%)
T1: PolyUHK	88	87	91	86
$\boldsymbol{C_{token_0}}$	83	84	84	86
$\boldsymbol{C_{token_5}}$	87	86	91	86

Table 5.5 Comparison between Our Token-based Approach and Top System using Purity Scores

Table 5.4 and Table 5.5 show that our approach $\boldsymbol{C_{token_5}}$ achieves a similar result with the *PolyUHK* system in BCubed and Purity F-scores when it uses contextual relevance weighting for the surrounding five tokens. In comparison to the $\boldsymbol{C_{token_0}}$ system, we found that $\boldsymbol{C_{token_5}}$ has obtained an increase of 6% in BCubed $F_{\alpha=0.5}$ and 4% in BCubed $F_{\alpha=0.2}$. $\boldsymbol{C_{token_5}}$ has also obtained an increase of 4% in Purity $F_{\alpha=0.5}$ and 2% in Purity $F_{\alpha=0.2}$. This demonstrates the effectiveness of our approach in capturing the information neighboring the person name mentions. Our approach can achieve a comparable result using tokens of nouns, verbs and adjectives. It is, however, restricted by high dimension of features, making the curse of dimensionality emerge as a great challenge. In the WePS2 data set, the number of tokens and keywords for each person is given in Figure 5.6.

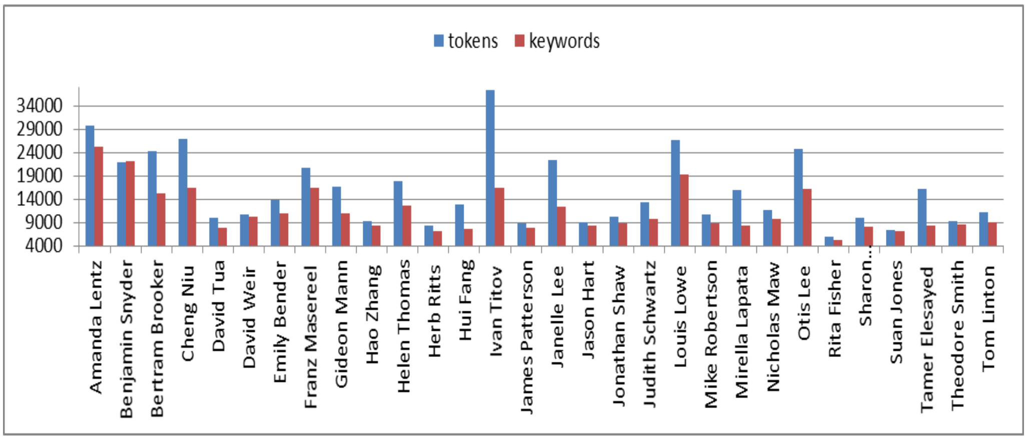


Figure 5.6 Number of Tokens and Keywords for 30 Persons

The horizontal axis represents person names in WePS2 dataset and the vertical axis shows the number of features (tokens and keywords). Figure 5.6 shows that the number of keywords is smaller than that of tokens for each person. The problem is whether the clustering effectiveness is guaranteed when using keywords with contextual relevance. We then compared $\boldsymbol{C_{keyword_3}}$ with $\boldsymbol{C_{token_5}}$ in Table 5.6 and Table 5.7. $\boldsymbol{C_{token_5}}$ obtains the optimal results when the threshold is set to 0.135. However, in the WePS2 workshop, the top performing system sets the threshold to 0.1. Therefore, we also present the results for $\boldsymbol{C_{tken_5}}$ by the threshold of 0.1.

Systems	**Macro-averaged Scores**			
	BCubed F-scores			
	α= 0.5 (%)	α= 0.2 (%)	BEP (%)	BER (%)
$\boldsymbol{C_{token_5}}$, threshold=0.1	75	82	69	91
$\boldsymbol{C_{token_5}}$, threshold=0.135	81	80	87	79
$\boldsymbol{C_{keyword_3}}$, threshold=0.1	**82**	**82**	85	82

Table 5.6 Comparison between Contextual Keywords and Tokens using BCubed Scores

Systems	**Macro-averaged Scores**			
	Purity F-scores			
	α= 0.5 (%)	α= 0.2 (%)	Purity (%)	InvPurity (%)
$\boldsymbol{C_{token_5}}$, threshold =0.1	82	88	76	94
$\boldsymbol{C_{token_5}}$, threshold =0.135	87	86	91	86
$\boldsymbol{C_{keyword_3}}$, threshold=0.1	**88**	**88**	89	**88**

Table 5.7 Comparison between Contextual Keywords and Tokens using Purity Scores

In comparison to $\boldsymbol{C_{token_5}}$ with threshold being equal to 0.135,

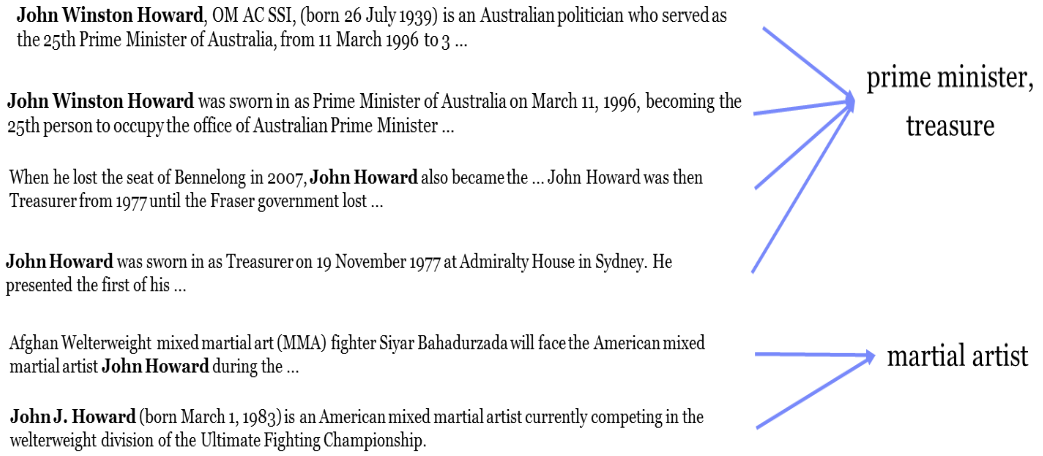
$\boldsymbol{C_{keyword_3}}$ gained 1% increase in both BCubed and Purity $F_{\alpha=0.5}$ and 2% increase in both BCubed and Purity $F_{\alpha=0.2}$. This shows that keywords can increase semantic relevance of words in WPD. Comparatively, using tokens with contextual relevance, our system achieves the optimal result when the window size is 5. This is because keywords are bigger in granularity than tokens. Meanwhile, it is obvious that the HAC algorithm is rather sensitive to the threshold. Compared to the 0.1 threshold value, $\boldsymbol{C_{token_5}}$ obtains a higher increase in precision at the cost of recall when the threshold is set to 0.135. This is why we investigate the co-reference technique for resolving person name ambiguities.

5.2 Hierarchical Co-reference Modeling for WPD

In Section 5.1 Contextual Relevance Weighting for WPD, we have evaluated the contextual relevance of keywords for person name disambiguation. Experimental results show that keywords are informative contextual clues to separate persons in the Hierarchical Agglomerative Clustering (HAC) algorithm. However, the HAC algorithm needs to manually tune the threshold for the number of clusters. In fact, there is no way to determine the threshold for certain, because the number of clusters changes from name to name. In this thesis, we propose using a semantic-based hierarchical co-reference resolution technique that does not require threshold tuning. The algorithm recursively partitions potential entities into a tree structure with latent sub-entities as child nodes and person names as observable leaf nodes. Person names are then disambiguated by deciding whether two entity nodes are co-referential or not. Experiments on the WePS2 dataset have shown that our approach, which is not dependent on the training data, has achieved a comparable performance with the top two state-of-the-art systems. In the following sections, we first introduce the pairwise co-reference modeling for entity disambiguation and then describe the hierarchical co-reference technique for WPD.

5.2.1 Pairwise Co-reference Modeling

Given a collection of name mentions extracted from document texts, co-reference resolution for WPD is to group name mentions into clusters such that two mentions in the same cluster refer to the same real-world entity. For example, the query "*John Howard*" can have the following returned search results,



John Winston Howard, OM AC SSI, (born 26 July 1939) is an Australian politician who served as the 25th Prime Minister of Australia, from 11 March 1996 to 3 ...

John Winston Howard was sworn in as Prime Minister of Australia on March 11, 1996, becoming the 25th person to occupy the office of Australian Prime Minister ...

When he lost the seat of Bennelong in 2007, **John Howard** also became the ... John Howard was then Treasurer from 1977 until the Fraser government lost ...

John Howard was sworn in as Treasurer on 19 November 1977 at Admiralty House in Sydney. He presented the first of his ...

Afghan Welterweight mixed martial art (MMA) fighter Siyar Bahadurzada will face the American mixed martial artist **John Howard** during the ...

John J. Howard (born March 1, 1983) is an American mixed martial artist currently competing in the welterweight division of the Ultimate Fighting Championship.

Figure 5.7 Example Mentions of "*John Howard*" with the True Entities

In Figure 5.7, for the query name "*John Howard*", we can have a list of returned search results with such name mentions as "*John Winston Howard*", "*John Howard*", "*John J. Howard*". The six name mentions refer to two real world entities: prime minister and martial artist. Additionally, the prime minister "*John Winston Howard*" was also the treasurer before he became prime minister.

Previous works tend to place name mentions into groups using clustering methods (Bagga and Baldwin, 1998; Gooi and Allan, 2004; Rao et al., 2010). Such methods requires tuning the number of clusters manually. In addition, the pairwise co-reference model has been proposed to solve this problem. This model creates a pairwise factor graph with factors measuring the compatibility between two name mentions (McCallum and Wellner, 2004). Take the name mentions in Figure 5.7 for example, a pairwise factor graph can be plotted as follows,

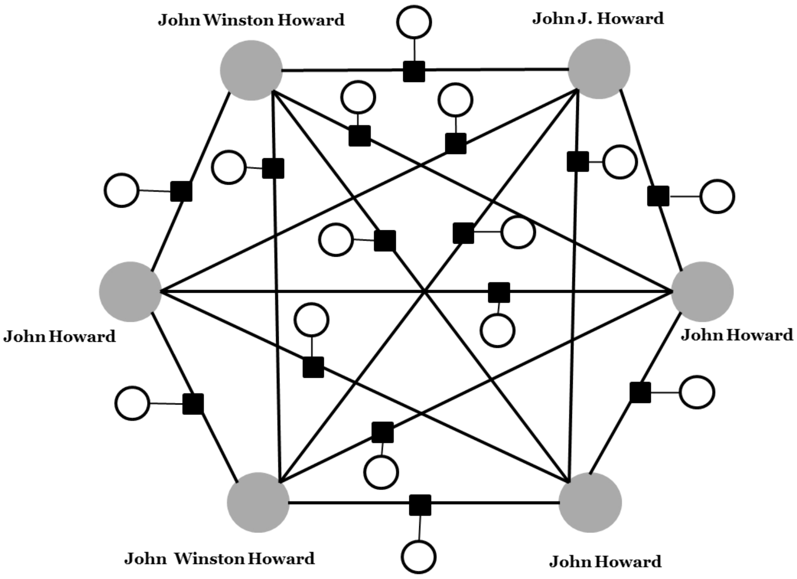


Figure 5.8 Pairwise Co-reference Model for the Query "*John Howard*"

In Figure 5.8, the open circles are binary co-reference decision variables indicating whether the two mentions are co-referent or not; shaded circles are observable name mentions and black boxes are factor nodes that represent the pairwise compatibility functions between name mentions. Given a set of mentions $\boldsymbol{m}$, let $\boldsymbol{y}$ be the set of decision variables where y_{ij} is a binary decision variable which is defined as,

$$y_{ij} = \begin{cases} 1, if\ m_i\ is\ coreferential\ to\ m_j \\ 0, otherwise \end{cases}$$

In the pairwise co-reference model, the probability of $\boldsymbol{y}$ given $\boldsymbol{m}$ is defined by,

$$p(\boldsymbol{y}|\boldsymbol{m}) \propto \prod_{i=1}^{n}\prod_{j=1}^{n}\varphi(m_i, m_j, y_{ij})$$

where φ is the factor that assesses the compatibility between mention pairs and outputs a score indicating how likely it is that a pair of mentions is referring to the same entity. The compatibility functions can be computed over the feature vectors that are derived from the surface texts of name mentions and the contexts surrounding the name mentions. Therefore, co-reference resolution is to search for an optimal setup of decision variables $\boldsymbol{y}$ with the highest probability. However, the pairwise co-reference model has a quadratic number of decision variables when there is a large number of name mentions. To solve this problem, Wick et al. (2012) proposed a hierarchical co-reference model which recursively partitions entities into a tree structure with latent sub-entities as child nodes and name mentions as observable leaf nodes. By means of the hierarchical tree structure, this model can accumulate different features from child nodes to enrich the feature space in the parent node. It can also scale up to a large collection of name mentions. For these reasons, we decided to use the hierarchical co-reference model for WPD with no need to manually tune the number of clusters.

5.2.2 Hierarchical Co-reference Modeling

The basic idea of the hierarchical co-reference modeling technique is to recursively partition entities into a tree structure with latent sub-entities as child nodes and name mentions as observable leaf nodes. Name mentions in Figure 5.7 can be organized into a tree with the factor nodes added between parent and child nodes:

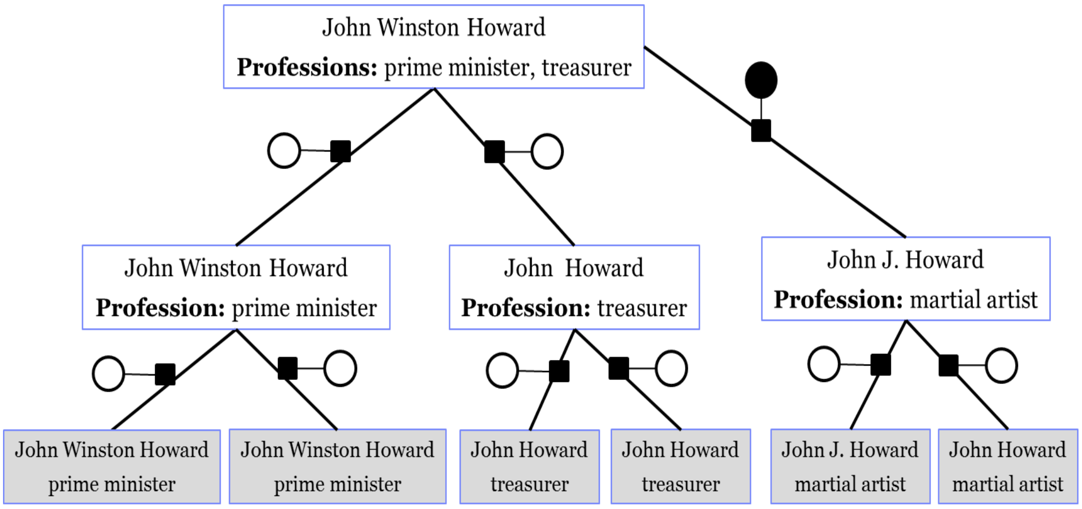


Figure 5.9 Hierarchical Co-reference Model for the Query "*John Howard*"

In the Figure 5.9, name mentions are observable leaf nodes in gray boxes. Decision variables in open circles express the parent-child relationship between two entity nodes. Factor nodes in black boxes measure the compatibility between parent and child nodes. Latent entity nodes in white boxes aggregate attributes from child nodes (sub-entity nodes or leaf nodes). The factor in the shaded black circle decides whether the two sub-trees are co-referent or not. It can be seen that the co-reference resolution is conducted between entity nodes instead of between the mention pairs, thus greatly reducing the number of decision variables while increasing the representation power of the entity nodes since the entity nodes aggregate the attributes from their child nodes.

Let e_i denote the i^{th} entity node and m_j be the j^{th} name mention in the tree. Let y_{ij} be a binary decision variable indicating whether m_j co-refers to the parent entity e_i. Formally, the probability distribution of the hierarchical co-reference model can be defined as,

$$p(\boldsymbol{y}, \boldsymbol{E}|\boldsymbol{m}) \propto \prod_{e_i \in E} \varphi_1(e_i)\varphi_2(e_i, e_i^p)$$

where $\boldsymbol{E}$ is the set of entity nodes. Factor φ_1 assesses the prior knowledge over the entity nodes, for example, the size of an entity node. Factor φ_2 measures the compatibility between a child entity node and its parent node denoted by e_i^p. In this formal representation, factors φ_2 can take the form of an exponential function $exp(\boldsymbol{\theta} \cdot \emptyset(e_i, e_i^p))$, $\emptyset(e_i, e_i^p)$ are feature functions for the entity node e_i and its parent. For example in WPD, the feature functions can test whether two entity nodes have the same emails or compute the cosine similarity between two entities' bag-of-words derived from their child nodes. The parameters $\boldsymbol{\theta}$ indicate the importance of these feature functions. To learn these parameters, the Markov chain Monte Carlo (MCMC) inference algorithm can be used to search for a configuration of the entity trees that that has the highest probability (Wick et al., 2012). In each MCMC step, it randomly selects two sub-trees with entity nodes e_i and e_j. Then merge and split operations are conducted in the following cases. Suppose e_i and e_j are the left and right sub-trees,

Case 1: If e_i and e_j are in the same cluster, then the following proposals are made,

- Split right. Detach e_j from the tree and make it an independent sub-tree.
- Collapse. If e_i has a parent node, attach e_i's children to its parent and remove the e_i node.
- Sample attribute. Sampling a new value for e_i's attribute from its child nodes.

Case 2: If e_i and e_j are in different clusters, then the following proposals are made,

- Merge left. Detach e_j from its parent node and make it as a child node to e_i.
- Merge entity left. Make e_j's parent node as a child to e_i.
- Merge left and collapse. Merge e_j into e_i and remove the e_j node.

- Merge up. Create a new parent node and attach e_i and e_j to the parent node.

To accept or reject these proposals, the Metropolis Hastings (MH) sampler is used (Culotta and McCallum, 2006, Wick et al., 2012). Based on the current co-reference configuration $\boldsymbol{y}$, a proposal function puts forward a new configuration $\boldsymbol{y}'$ by the merge and split operations. These proposed changes are accepted with the probability α which is defined as,

$$\alpha(\boldsymbol{y}',\boldsymbol{y}) = min\left(1,\frac{p(\boldsymbol{y}')}{p(\boldsymbol{y})}\frac{q(\boldsymbol{y}|\boldsymbol{y}')}{q(\boldsymbol{y}'|\boldsymbol{y})}\right)$$

where *q* is a transition kernel. MH sampler is a special case of the Markov chain, and if the chain is reversible, then $q(\boldsymbol{y}|\boldsymbol{y}') = q(\boldsymbol{y}'|\boldsymbol{y})$, thus the acceptance probability α is reduced to,

$$\alpha(\boldsymbol{y}',\boldsymbol{y}) = min\left(1,\frac{p(\boldsymbol{y}')}{p(\boldsymbol{y})}\right)$$

which simply measures the model ratio between the current co-reference configuration $\boldsymbol{y}$ and the proposed configuration $\boldsymbol{y}'$.

After introducing the formal definitions, the next step for hierarchical co-reference is to select features from which the factors (compatibility functions) can be computed. In WPD, the name mentions are often extracted from Web text, and the context surrounding the name mentions is informative in identifying a person. This has already been proved by experiments in 5.1 Contextual Relevance Weighting for WPD. The **context** of a name mention is defined to be all the words inside a specified window around the name mention. In this thesis, the window size is set to 55 words, a parameter experimentally determined to give optimum performance in the task of cross-document co-reference resolution (Gooi and Allan, 2004). When the context words of the name mentions are overlapped, the overlapped words are extracted only once. Within this context, we devise a set of semantic features suitable for co-reference resolution as given below:

(1) Local words. All the words insides a specified window size will be used. In this thesis, nouns, adjectives and verbs are used.

(2) HTML features. HTML documents offer many structured information using different tags, for example, "*href*" denotes the destination link; "*b*" and "*i*" emphasize the content in bold and italic style; "*H1*", "*H2*" are for header elements. "*title*" represents the title of an HTML document. Therefore, we explore the following HTML features within the context of name mentions.

(2.a) **Titles**. Title tokens are used if they are in the neighborhood of a name mention.

(2.b) **Snippets**. Snippets are a summarization of HTML documents. Snippet tokens are used if they are located within the name mention's contexts.

(2.c) **URLs.** URLs are extracted. From these URLs, we also extract the hosts and anchors using a pattern-based approach. In this thesis, we have used URLs from the "*href*" and "*img*" tags.

(2.d) **Bold texts**. Bold texts are extracted from the tag "*b*". Also, texts from the "*H1-H4*" are placed under this category.

(2.e) **Italic texts**. Italic texts are extracted from the tag "*i*".

(3) Person-specific features: They include full name, email, profession, dates, ages, phone numbers and genders. With the exception of profession, the other six features are extracted based on rules. Professions are extracted by searching the manually crafted profession dictionary. Designators for genders are *Mr., Mrs., Miss, Ms., Lady, Lord* and so on. We also use a rule-based gender extractor mostly based on the patterns given in (Bergsma et al., 2009; Ji and Lin, 2009). If a sentence containing

the target person name meets the following pattern,

Gender Pattern	**Examples**
NN + and/or + his/her	Benjamin Snyder and his wife … Helen Thomas and her peers …
he/she + VB + DT + NN	Tua says he needs a knockout She is the author of the booklet … Jonathan Shaw
NN + VB + he/she	Sharon Cummings said she created Deja vu … David Tua showed he is on the way ….
NN + VB + his/her	Amanda Lentz defended her PhD thesis … Benjamin Snyder left his start …
NN + VB + himself/herself	Helen Thomas is herself the story … Tom Linton roused himself from a chilly doze …

Table 5.8 Gender Patterns and Corresponding Examples

A masculine or feminine value will be assigned to the target person.

(4) Surrounding named entities and types: Persons, locations and organizations within the context of a name mention will be used. The entity types will be added as features as well.

(5) Keywords: They are informative for a person if they are located within the context of the person name mention, for example, keywords “*tv series*”, “*sound clip*” and “*imdb*” can identify a person as an *actor*, whereas “*rowlings*” and “*trampoline*” define a person as an athlete. In this thesis, keywords are extracted using the CRFs model with the naturally annotated resource described in Chapter 4.

(5) Wikipedia categories: To enrich the feature representation of target person names, we extract Wikipedia categories for the keywords, named entities and bold texts in the proximity of the name mentions. For example, the name mention “*Amanda Lentz*” has a neighboring named entity “*North Carolina*” which contains the following category labels in Wikipedia:

Former British colonies
Spanish colonization of the Americas
State of Franklin
States and territories established in 1789
States of the Confederate States of America
States of the United States
North Carolina
Southern United States

Table 5.9 Categories for "*North Carolina*" in Wikipedia

These categories will be used as features.

(6) Topics. We apply the topic modeling technique to find the topics for each name mention (Blei et al., 2003). This is because we believe that if two name mentions refer to the same underlying entity, the context surrounding them is supposed to have the same topics. Each document for a name mention is represented by a sequence of topic distributions that can be plotted as,

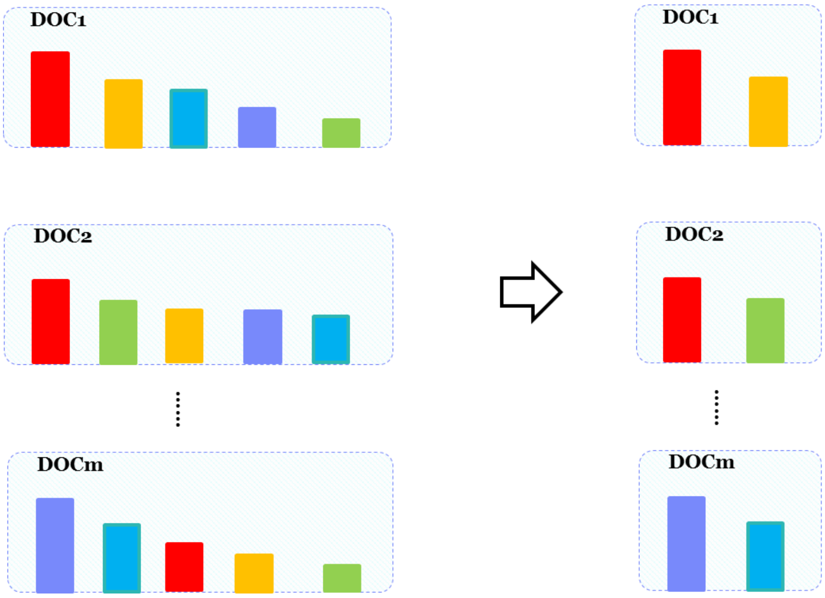


Figure 5.10 Topic Distributions for *m* Documents

Suppose we have *m* documents, we extract five topics for each document. The topics within documents are sorted by their probability distributions. Figure 5.10

shows the sorted topics for each document and two topics with the highest probabilities are selected. These selected topics will be used as features in WPD.

These seven types of features are represented by the bag-of-words model. Factors are then defined over these features. In this hierarchical co-reference model, six types of factors are used. They are,

(1) Bag-of-words cosine similarity. This factor examines the compatibility between parent bag and child bag. It is defined by,

$$w \times log(\|\boldsymbol{c}\|_1 + 2)\left(\frac{(\boldsymbol{p} - \boldsymbol{c}) \cdot \boldsymbol{c}}{\|\boldsymbol{p} - \boldsymbol{c}\|_2 \|\boldsymbol{c}\|_2} + t\right)$$

where w and t are weights to be tuned. $\boldsymbol{p}$ and $\boldsymbol{c}$ refer to the parent bag and child bag. $(\boldsymbol{p} - \boldsymbol{c})$ is used to remove the double counting of the features in the parent bag $\boldsymbol{p}$ from its child bag $\boldsymbol{c}$. $\|\boldsymbol{c}\|_1$ and $\|\boldsymbol{c}\|_2$ are l_1 and l_2 norms of the child bag. Formally the l_n-norm of a bag is given by,

$$\|\boldsymbol{b}\|_n = \sqrt[n]{\sum_{x_i \in \boldsymbol{b}} |x_i|^n}$$

where x_i is the weight of the i^{th} term. $\frac{(\boldsymbol{p}-\boldsymbol{c})\cdot\boldsymbol{c}}{\|\boldsymbol{p}-\boldsymbol{c}\|_2\|\boldsymbol{c}\|_2}$ is the cosine similarity between parent and child bags.

(2) Bag-of-words entropy penalty. This factor assesses the entropy of a bag of features. It is defined as,

$$-w \times \frac{H(\boldsymbol{b})}{log\|\boldsymbol{b}\|_0}$$

where $\boldsymbol{b}$ is a bag of features, and $H(\boldsymbol{b})$ is the entropy of the bag. $\|\boldsymbol{b}\|_0$ is l_0-norm of the bag which normalizes entropy to the range [0,1]. The entropy measures the amount of information carried by a term x. $H(\boldsymbol{b})$ is defined as,

$$H(\boldsymbol{b}) = -\sum_{x_i \in \boldsymbol{b}} \frac{x_i}{\|\boldsymbol{b}\|_1} \log\left(\frac{x_i}{\|\boldsymbol{b}\|_1}\right)$$

where x_i is the weight of i^{th} feature. This factor tends to penalize bags having higher entropy than those having lower entropy. For example, in WPD, we expect a person to have primary keywords "*tv series*" and "*imdb*" (lower entropy bag), and we would penalize a co-reference resolution in which we predict a person with a dozen keywords (higher entropy bag, for example, "*tv series*", "*imdb*", "*rowlings*" and "*trampoline*").

(3) Bag-of-words complexity penalty. This factor measures the capability of a bag of features. It is defined by,

$$-w \times \frac{\|\boldsymbol{b}\|_0}{\|\boldsymbol{b}\|_1}$$

where $\|\boldsymbol{b}\|_0$ counts the number of non-zero elements in a bag of features. This factor tends to discover entities that have primary frequent features (lower entropy bag).

(4) Entity existence penalty. This factor examines whether the node e is the root in the tree or not. It is defined by,

$$-w \times f(e)$$

where w is weight and $f(e)$ is an indicator function and returns 1 if the node e is the root and 0 otherwise.

(5) Sub-entity existence penalty. This factor examines whether the node e is a root or a leaf. It is defined by,

$$-w \times g(e)$$

where w is weight and $g(e)$ is an indicator function and returns 1 if the node e is neither a root nor a leaf in the tree and 0 otherwise.

(6) Name penalty. This factor measures the compatibility between name mentions in the bag. It is defined as,

$$-min\big((w \times \|\boldsymbol{b}\|_0 - 1), -t\big)$$

where t is a saturation parameter that imposes the lower limit on the name penalty.

5.2.3 Performance Evaluation

The evaluation of the hierarchical co-reference algorithm for WPD is conducted using the test data of WePS2 workshop 2009 which has 30 ambiguous names with two sets of evaluation metrics: the BCubed precision and recall (BEP and BER in short), Purity and Inverse Purity (InvPurity in short) (Artiles et al. 2007, 2009). Two F-scores are also used, one giving equal weighting to precision and recall (α=0.5) and the other giving higher weighting to recall (α=0.2). In this work, we use the Wikipedia dump with the timestamp: April 03, 2013 for categories and keyword extraction. The Stanford CoreNLP tool is used to preprocess these articles, including tokenization, part-of-speech tagging, and named entity recognition. For named entities, we treat consecutive tokens with the same category as a single entity mention. For topics of each name mention, we run a topic modeling procedure with hyper-parameters α=0.1 and β=0.1 by 100 iterations. The number of topics is set to 50 for each person name and 10 topics with highest probabilities are kept for each document.

In this thesis, there are six types of factors: cosine similarity, entropy, complexity, names penalty, entity existence penalty, and sub-entity existence penalty. Each factor has one or two parameters and the configuration of parameters is given in Table 5.10.

Features	**Weights for Factors**
keywords emails neighbor categories	$w = 4.0, t = -0.25\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
italics entity types ages snippet phone numbers professions dates bold text categories keyword categories	$w = 4.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
local words topics	$w = 4.0, t = -0.25\ (cosine)$ $w = 0.75\ (entropy)$ $w = 0.25\ (complexity)$
neighboring named entities URLs	$w = 2.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
bold texts title tokens destination URL titles	$w = 3.0, t = -0.125\ (cosine)$ $w = 0.25\ (entropy)$ $w = 0.25\ (complexity)$
middle names gender	$w = 1.0, t = 16\ (name\ penalty)$
structural prior	$w = 4.0\ (entity\ existence\ penalty)$ $w = 0.25\ (subentity\ existence\ penalty)$

Table 5.10 Weights for Factors in the Hierarchical Co-reference Model

For all features used by Wick et al. (2012) for author co-reference, we simply follow their weight configurations. These features are topics and names. For the new features in this thesis, we also use the weight configurations which can find corresponding characteristics in Table 5.10.

Our system $HIER_{coref}$ is first compared to four of the systems in the WePS2 workshop which have tuned the threshold automatically. The *XMEDIA_3* system estimated the threshold from the training data using the Quality Threshold clustering algorithm (Romano et al., 2009). The *UMD_4* system learned the threshold for the HAC by the Support Vector Machines (Gong and Oard, 2009). The *CSSIANED_4* system is based on the professional category information to disambiguate namesakes by categorizing them into a professional taxonomy from the Freebase using the KNN classifier (Han and Zhao, 2009). The *PRIYAVEN* system applied the Fuzzy Ant Clustering which works without specifying the number of clusters (Venkateshan, 2009). The comparisons are given in Table 5.11 and Table 5.12.

Systems	**Macro-averaged Scores**			
	BCubed F-scores			
	α= 0.5 (%)	α= 0.2 (%)	BEP (%)	BER (%)
XMEDIA_3	72	68	82	66
UMD_4	70	63	**94**	60
CASIANED_4	63	68	65	**75**
PRIYAVEN	39	37	61	38
$\boldsymbol{HIER_{coref}}$	**81**	**78**	88	**78**

Table 5.11 Comparison in BCubed Scores between Our $HIER_{coref}$ System and Other Systems that Automatically Tune Threshold

Systems	**Macro-averaged Scores**			
	Purity F-scores			
	α= 0.5 (%)	α= 0.2 (%)	Purity (%)	InvPurity (%)
XMEDIA_3	80	76	91	73
UMD_4	81	76	**95**	72
CASIANED_4	73	77	72	83
PRIYAVEN	73	77	72	83
$\boldsymbol{HIER_{coref}}$	**86**	**84**	91	**83**

Table 5.12 Comparison in Purity Scores between Our $HIER_{coref}$ System and Other Systems that Automatically Tune Threshold

Table 5.11 and Table 5.12 show that $HIER_{coref}$ obtains the highest F-scores in both BCubed and Purity scores. When compared to the *XMEDIA_3* system, $HIER_{coref}$ obtains 9% and 10% increase in BCubed $F_{0.5}$ and $F_{0.2}$; 5% and 8% increase in Purity $F_{0.5}$ and $F_{0.2}$. Similar to the single pass clustering, *XMEDIA_3* applied the Quality Threshold clustering algorithm, the threshold for merging two documents are learned using the SVM regression model. This implies that they need training data for their learning model. As with the *XMEDIA_3* system, the *UMD_4* system learned the threshold for the HAC by the Support Vector Machines. The *CSSIANED_4* system disambiguates person names by categorizing them into a professional taxonomy from Freebase using the KNN classifier. However, using professional categories to classify name mentions into different clusters has its own problem when a person has more than one profession. Examples in Figure 5.7 show that the entity "*John Howard*" has two professions: prime minister and treasurer in different time periods.

The next experiment compares $HIER_{coref}$ with the top four systems in the WePS2 workshop. They are denoted by *T1: PolyUHK* (Chen et al. 2009), *T2: UVA_1* (Balog et al. 2009) and *T3: ITC_UT_1* (Ikeda et al. 2009), T4:*XMEDIA_3* (Romano et al., 2009) and T4: *UMD_4* (Gong and Oard, 2009). The comparisons using BCubed and Purity scores are given in Table 5.13 and Table 5.14.

Systems	**Macro-averaged Scores**			
	BCubed F-scores			
	α= 0.5 (%)	α= 0.2 (%)	BEP (%)	BER (%)
T1: PolyUHK	82	80	87	79
T2: UVA_1	81	80	85	80
T3: ITC_UT_1	81	76	93	73
T4: XMEDIA_3	72	68	82	66
$\boldsymbol{HIER_{coref}}$	81	78	88	78

Table 5.13 Comparison in BCubed Scores with the Top Four Systems

Systems	**Macro-averaged Scores**			
	Purity F-scores			
	α= 0.5 (%)	α= 0.2 (%)	Purity (%)	InvPurity (%)
T1: PolyUHK	88	87	91	86
T2: UVA_1	87	87	89	87
T3: ITC_UT_1	87	83	95	81
T4: UMD_4	81	76	95	72
$\boldsymbol{HIER_{coref}}$	86	84	91	83

Table 5.14 Comparison in Purity Scores with the Top Four Systems

Table 5.13 and Table 5.14 show that $HIER_{coref}$ achieves a comparable performance with top two systems *PolyUHK* and *UVA_1* in both BCubed and Purity F-scores. It is worth noting that the top two systems each use the HAC algorithm, which requires tuning the threshold for determining the number of clusters. This is particularly challenging for WPD, since we do not know beforehand how many clusters exist in the search results for a given person name. Moreover, the number of clusters returned by the HAC algorithm is quite sensitive to thresholds, especially when the number of clusters per person name has a large variability among the 30 persons (from 1 up to 56 different persons sharing the same name). A tiny variation in the thresholds can significantly change the performance of the WPD system, which can be seen in experiments for token features in Section 5.1 Contextual Relevance Weighting for WPD.

5.3 Further Analysis of WPD

5.3.1 Validating Knowledge Base using WPD

As many people are developing knowledge bases which contain entity relationship information such as Freebase, Google Knowledge Graph, and Microsoft

Bing Satori, a natural question is that how good are the coverage of these knowledge bases. In other words, would WPD be useful in the world of ever growing knowledge base construction. To answer this question, we have conducted a coverage analysis of WePS2 in a publically accessible knowledge base Freebase (Google Knowledge Graph is not available for use). For this purpose, we use the Freebase snapshot of June 23, 2013 as the knowledge base. Freebase is a community-curated database that contains facts about named entities and their relations. Out of the 30 ambiguous names in the WePS2 dataset, 20 names indeed appeared in Freebase, giving a 66.67% of name coverage. However, this does not imply the coverage of the entities contained in Freebase is also 66.67%. To further identify how many entities given in WePS2 is covered by Freebase, we did a manual check to Freebase to find the number of Freebase entities and number of entities of the 30 names in the WePS2 web documents. Details can be found in Appendix 1. Figure 5.11 shows the summary of the result.

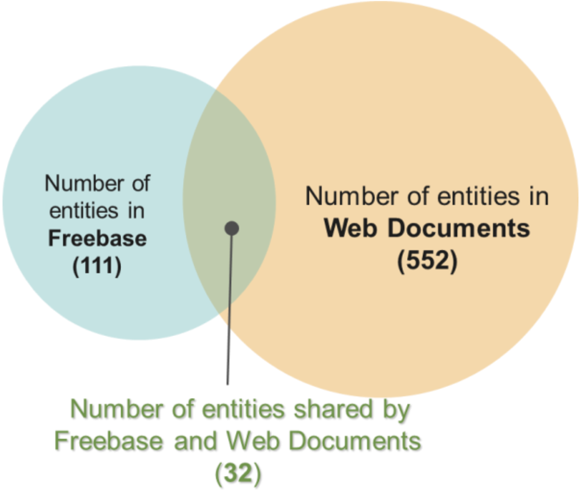

Number of entities in Freebase (111)
Number of entities in Web Documents (552)
Number of entities shared by Freebase and Web Documents (32)

Figure 5.11 Number of Entities in Freebase and WePS2 Dataset

Out of the 552 entities in WePS2 for the 30 names, only 32 entities are contained in Freebase, resulting in about 5.8% coverage to the WePS2 dataset.

Note that Freebase is built on such data sources as Wikipedia, EDGAR[7], Open Library Project[8], Stanford University Library[9], TVRage[10], MusicBrainz[11]. This 5.8% coverage is a good indication that knowledge base itself is insufficient for WPD. An automated WPD system is quite useful. It is interesting to note that Freebase does contain 79 entities out of the 30 names that are not in WePS data, contributing to about 14.3% more entities. This, on the other hand, is a good indication that knowledge base extracted from various resource, can contribute to about 15-20% of knowledge compared to using search engine results on which WPD method can be applied.

5.3.2 Complexity of Hierarchical Co-reference Model

To group web documents into different clusters with each cluster referring to the same real-world entity, hierarchical co-reference model recursively partition mentions/entities into a tree structure. This model has these advantages: a small number of upper-level entity nodes can summarize a large number of name mentions, increasing the model's representation power and its scalability over large collections of name mentions because co-reference decisions can be made between two entity nodes instead of mention pairs. However, it cannot handle the scenario when the two persons have the same set of features, but they do refer to different entities.

Since the hierarchical co-reference model is a randomized algorithm, it is

[7] http://www.sec.gov/edgar.shtml#.U__hVPmSwn4

[8] https://openlibrary.org/

[9] http://library.stanford.edu/

[10] http://www.tvrage.com/

[11] https://musicbrainz.org/

difficulty to evaluate its complexity directly. One practical way to estimate its complexity is by running experiments using different number documents. In the WePS2 data there are 7 names whose corresponding set of documents are below 100. So, in the complexity analysis, only 23 names are used. As HCM is a randomized algorithm, it requires a good sample size and the sample size depends the number of files to be clustered. Therefore, we need to obtain the appropriate sample size for different document set sizes. Figure 5.12 to Figure 5.15 give the performance of the algorithm after certain number of iterations (indicated by time here) for document size *N*=25, 50, 75, 100, respectively. Once the performance is stabilized, there is no point to run the sampling algorithm anymore.

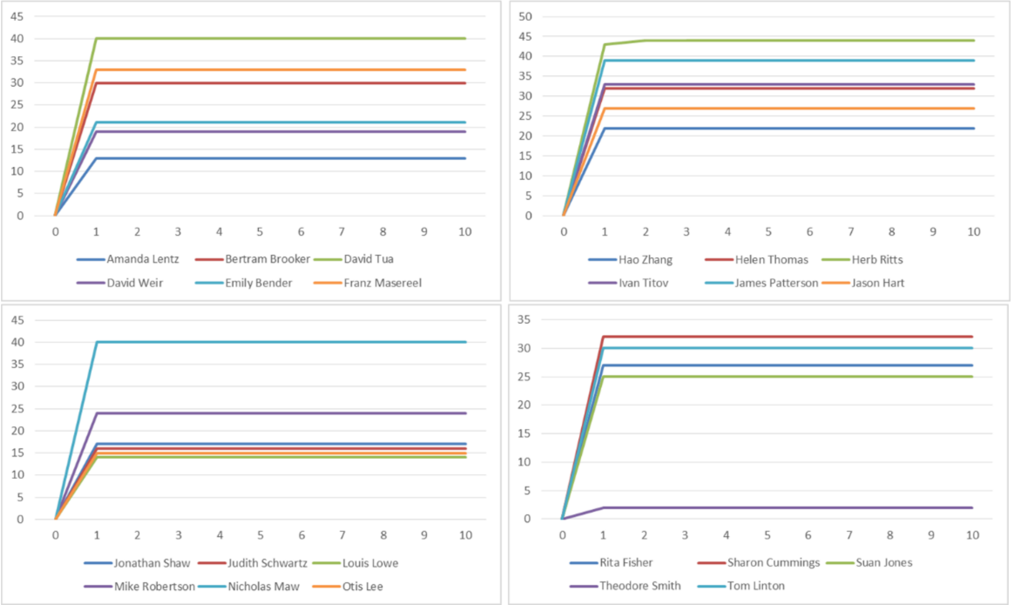


Figure 5.12 Sampling Performance for 23 Ambiguous Person Names with Each Person Name having 25 Documents

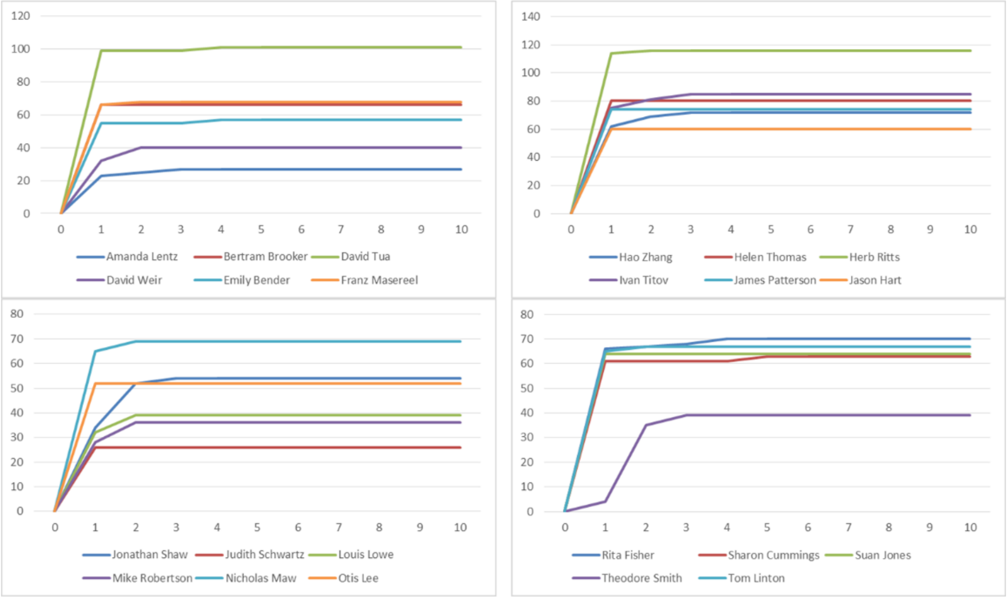


Figure 5.13 Sampling Performance for 23 Ambiguous Person Names with Each Person Name having 50 Documents

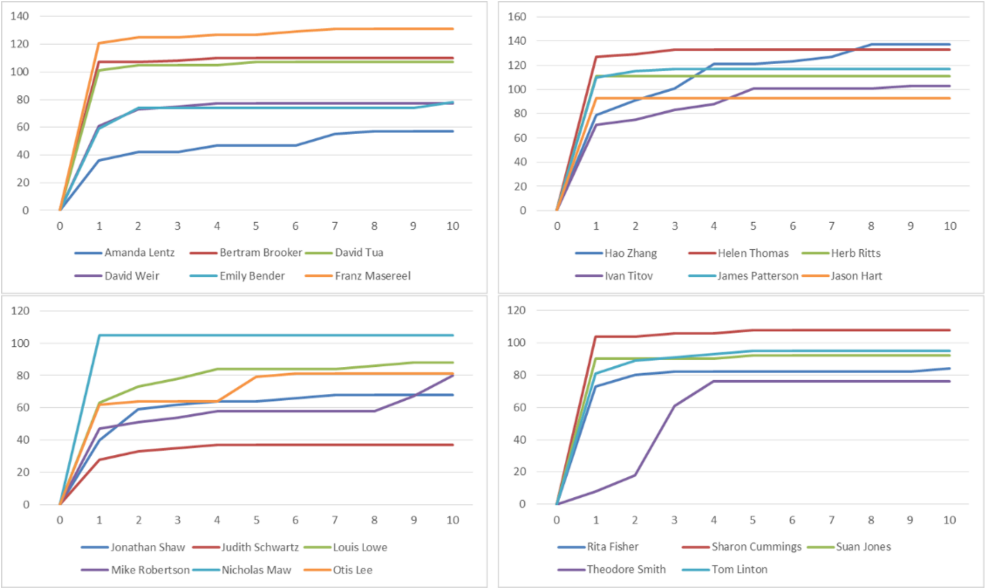


Figure 5.14 Sampling Performance for 23 Ambiguous Person Names with Each Person Name having 75 Documents

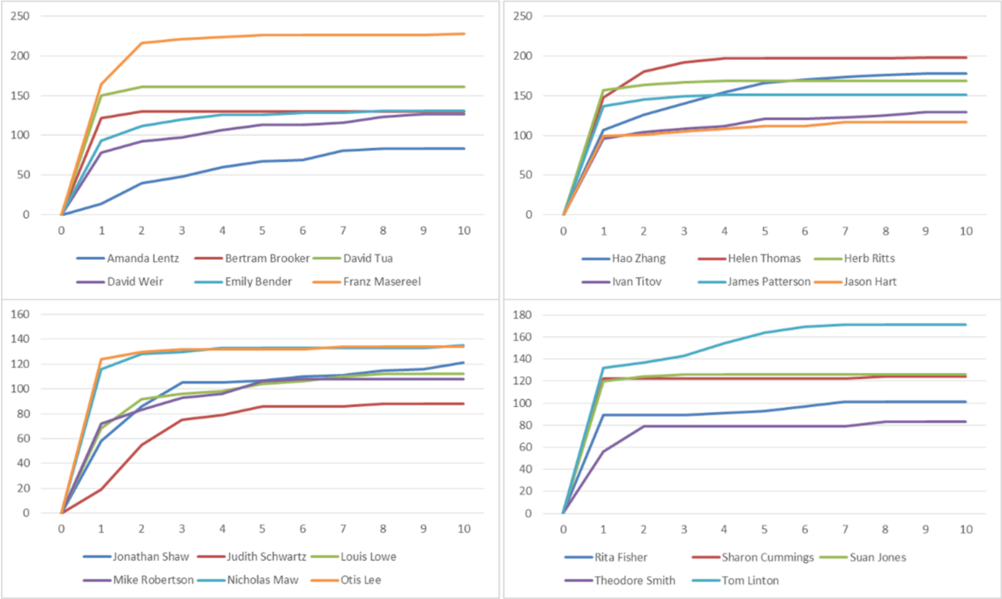


Figure 5.15 Sampling Performance for 23 Ambiguous Person Names with Each Person Name having 100 Documents

Figure 5.16 shows the relationship between the document size to the number of samples where the HCM has stabilized.

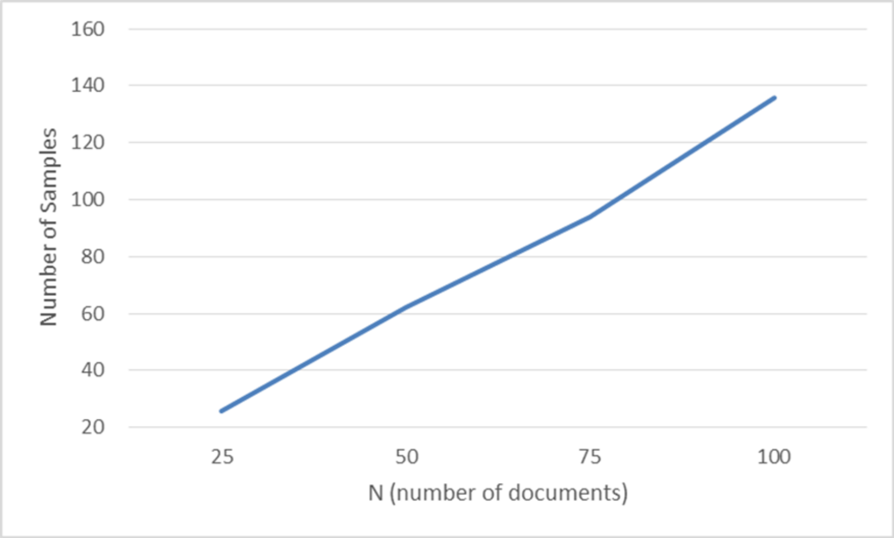


Figure 5.16 Relationship between Number of Documents and Sample Size

Roughly speaking, the relationship between the sample size and the number of documents is linear as shown in Figure 5.16. This is true at least when the number of documents is less than 100. In practice, people will hardly go beyond 100 documents to search for a person. So, the automated method is practical for use. In contrast, using pairwise method requires a quadratic comparisons of mentions ($O(N^2)$ in HAC algorithm (Sibson, 1973)). In addition, when proposing to merge or split a tree in the hierarchical co-reference model, the cost of evaluating the proposal depends on smaller number of factors.

5.4 Chapter Summary

This chapter proposes a simple but effective method for using single features with contextual relevance for Web Person Disambiguation (WPD). Experimental results show that contextual relevance can improve the overall performance of WPD in precision, recall and F-score. When tokens (nouns, verbs and adjectives) are used as features, the curse of dimensionality emerges as a great challenge. To resolve this problem, we explored the use of keywords as features for document representation. Experiments demonstrate the advantage of using keywords as features in dimensionality reduction and performance improvement in WPD. In terms of the clustering method, the HAC algorithm is used. It requires tuning a threshold to find the number of clusters, which is a particularly challenging problem of WPD, because we do not know beforehand how many clusters there are in the search results for a given person name. Worse still, the clustering solutions are rather sensitive to the tuned thresholds.

In this thesis, we have proposed using a hierarchical co-reference resolution technique that does not need to tune the threshold for determining the cluster number. Our disambiguation method is semantic-based and training data independent. Instead of using profession categories to determine the number of persons in the search results, we use professions only as features in the hierarchical co-reference model.

Experiments conducted on the WePS2 dataset show that the proposed method outperforms all systems that learn the threshold automatically. It also achieves a comparable performance with the top two systems which manually tune the number of clusters.

One important advantage of our disambiguation method is that it is semantic-based and training data independent. Another advantage of our approach is its scalability, because it can reduce the size of pairwise decisions between mentions due to its hierarchical organization of name mentions. This is particularly important for disambiguating millions of name mentions. However, our selection of person-specific features is rule-based. We plan to explore the use of supervised methods to extract these semantic features to augment WPD.

Chapter 6 Personal Profile Extraction

Once web pages are clustered into different targeted entities, information extraction is still needed to obtain each person's profiles such as birth date, children, sibling, education etc., as relational facts. Extracting personal profiles for a particular person enables users to uniquely identify that person on the web. For example, although different people might share the same name, they usually have different dates of birth or affiliations. Given a web document, the objective of personal profile extraction is to extract a pre-defined set of attribute values for a given person name. This process is often called **relation extraction**. A simple yet effective heuristic method is to use a set of seed examples or hand-written extraction patterns (Brin, 1998; Agichtein and Gravano, 2000; Etzioni et al., 2005; Banko et al., 2007). The benefit of this method is that no annotated data is needed, but the creation of reliable extraction patterns requires a great deal of expertise. Supervised learning methods are effective in learning personal relations with manually annotated data, such as tree kernel methods and maximum entropy model (Zelenko et al., 2003; Kambhatla, 2004; Culotta et al., 2006; Bunescu and Mooney, 2005a, 2006). However, these supervised methods rely heavily on annotated training data which is costly to yield. On a separate note, the distant supervision approaches without annotated data are particularly attractive because supervised relation extractors can be learned from voluminous facts in the existing knowledge base, Freebase in particular (Mintz et al., 2009; Riedel et al., 2010; Surdeanu et al., 2010, 2011; Hoffmann et al., 2011; Sun et al., 2011). Freebase, a community-curated database, contains facts about named entities and their relations. It is often used as the knowledge base to automatically generate labeled training data from Wikipedia for relation extraction.

Selecting training data based on Freebase for the current distance supervision algorithm can introduce noisy labeled data because they take all sentences mentioning two entities in a relation type as positive examples for a given relation. To improve the quality of selected training data, we consider two additional semantic constraints for relation extraction in a graphical model. The first is that two entities in question should have matching entity types for a given relation type. For instance, for the *birthplace* relation type, the first entity should be a Person and the second entity should be a Location. The second semantic constraint is for a given relation type, there must be some trigger words that are semantically relevant to it. For example, the surface form "*born*" certainly serves as a trigger word for the *birthplace* relation. In so doing, our semantic-based approach directly associates trigger words and entity types with relations in a graphical model. We learn the semantic trigger words automatically using a topic modeling approach (Blei et al., 2003).

6.1 Algorithm Design for PPE

In this work, relations extracted are of predefined types. For example, the *person.birthplace(Person, Location)* relation denotes a *Person* was born in a *Location*. The *organization.founders(Organization, Person)* relation refers to the fact that a *Person* is a founder of an *Organization*. *Kill(Person$_1$, Person$_2$)* indicates a *Person$_1$* killed/murdered *Person$_2$*. Also, any named entity must belong to one of the three **entity types**: Person, Location and Organization.

The goal of our algorithm is to include two semantic constraints, trigger words and entity types in a graphical model. To explain our model, we first present the two semantic constraints. Then we will explain how they can be integrated into the graphical model. Afterwards, a brief discussion of features will be presented.

6.1.1 Extraction of Trigger Words

Through investigation, we found that two entities coexisting in a sentence may not express a relation even if they do have one. For example,

> ***John/Clough/Holmes****, September/25/,/1809 -- December/16/,/1887, was responsible for the establishment of* ***Michigan/State/University*** *.*

We cannot identify the *employment_tenure* relation between "***John/Clough/Holmes***" and "***Michigan/State/University***" although the two entities indeed share the *employment_tenure* relation in Freebase. However, if lexical words such as, "*CEO*", "*professor*", "*managing director*" or "*dean*", *etc.* are found in such sentences, it is more likely that the *employment_tenure* relation exist between the entity pairs (person, organization).

To learn these semantically relevant trigger words for certain target relation types, we apply the topic modeling method--LDA (Blei et al., 2003). LDA is a generative model based on probabilistic sampling techniques investigating how words in documents are generated with hidden variables (Steyvers and Griffiths, 2006). Its main idea is to model documents in terms of topics where a topic is defined as a distribution over a fixed vocabulary of words. In this model, words in documents are observable variables and topics are latent variables hidden in these documents. Its graphical representation is given in Figure 6.1.

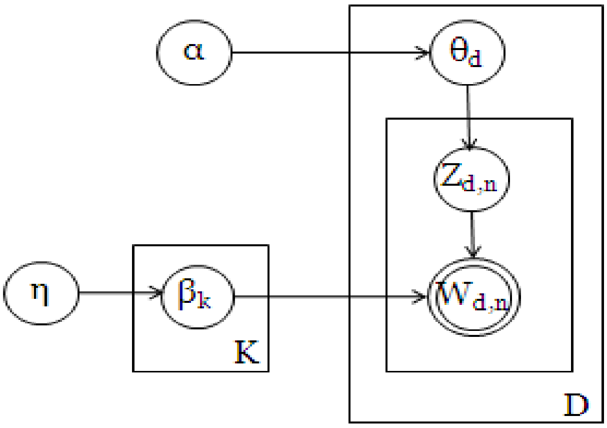


Figure 6.1 LDA Graphical Representation (Blei et al., 2003)

In Figure 6.1, each node denotes a random variable and the edge between nodes represents dependency relations between nodes. The double circles around the random variable denote an observable node (evidence node). The plate surrounding the nodes indicates *N* independent and identically distributed (i.i.d) samples. *D* and *K* refer to the number of documents and the number of topics, respectively. α and η are hyper-parameters on the mixture proportions for topics and documents. θ_d refers to the multinomial topic distributions for document *d* and β_k is multinomial word distributions for topic *k*. $Z_{d,n}$ denotes a topic from which the n^{th} word in document *d* is drawn and $W_{d,n}$ indicates the observable n^{th} word in *d*.

In the LDA model, for a document *d*, a vector of topic distributions $\overrightarrow{\theta_d}$ is drawn from a Dirchlet distribution $\overrightarrow{\theta_d} \sim Dir(\vec{\alpha})$; topic assignment for n^{th} word $Z_{d,n}$ follows from a multinomial distribution $\overrightarrow{Z_{d,n}} \sim Mult(\overrightarrow{\theta_d})$; and the n^{th} word $W_{d,n}$ in document *d* is sampled from multinomial distribution $\overrightarrow{W_{d,n}} \sim Mult(\overrightarrow{\beta_{Z_{d,n}}})$. To find topic distributions for each document, *p(z|w)* must be obtained for the hyper-parameters α and η. Since exact inference of this distribution is intractable, Gibbs sampler is used. When *p(z|w)* is obtained, the topic distributions θ for each document can be estimated. We then find the best topic that has the highest probability for each document and obtain the words under that best topic.

However, the problem is that each relation instance is represented by a sentence that is shorter than a document in size. The standard topic model captures a document-level word co-occurrence pattern to find topic distributions and would suffer from a data sparsity issue in sentences. To solve this problem, we randomly split sentences of a relation type into subsets and aggregate each subset of sentences into a document. In so doing, we can create a list of documents containing a number of sentences. On the basis of these documents, topic models will be trained to find the best topic for each document. The most frequent best topic will be selected for the target relation. The flow can be plotted as follows,

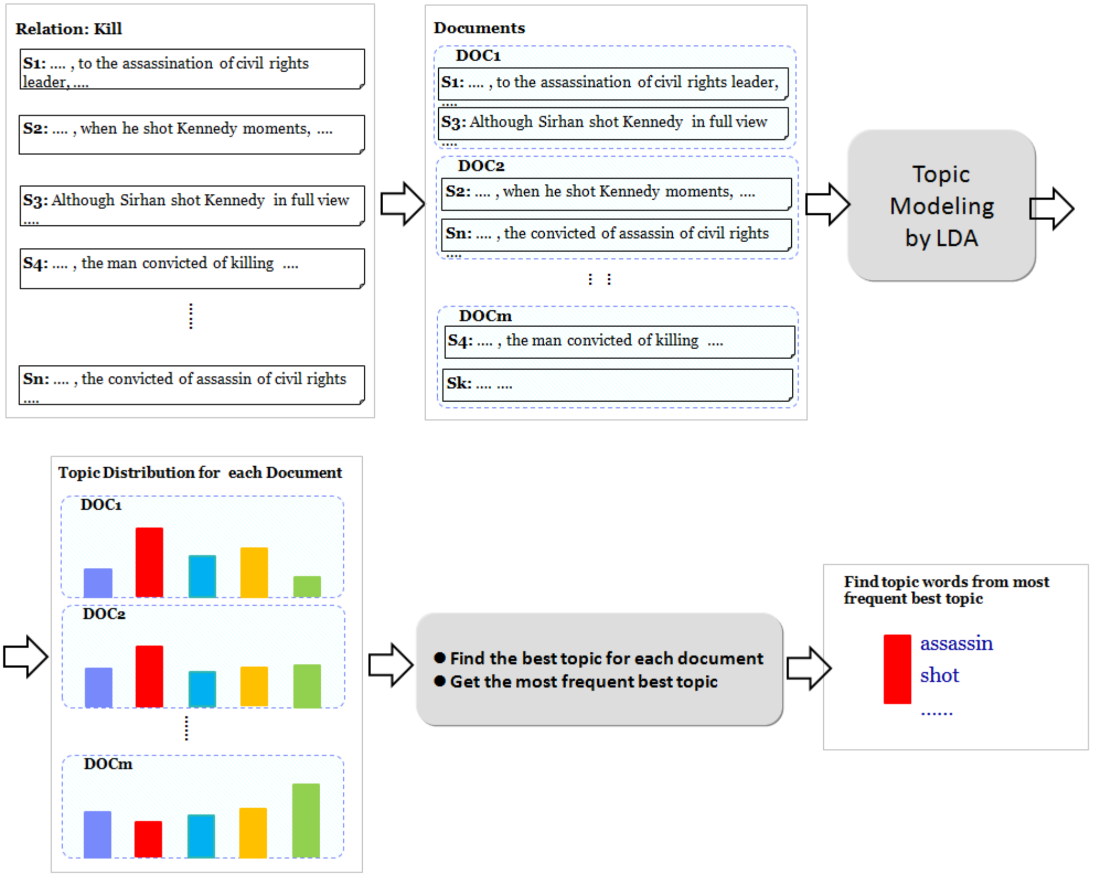

Relation: Kill
S1: .... , to the assassination of civil rights leader, ...
S2: .... , when he shot Kennedy moments, ...
S3: Although Sirhan shot Kennedy in full view ....
S4: .... , the man convicted of killing ....
Sn: .... , the convicted of assassin of civil rights ....
Documents
DOC1
S1: .... , to the assassination of civil rights leader, ...
S3: Although Sirhan shot Kennedy in full view ....
DOC2
S2: .... , when he shot Kennedy moments, ...
Sn: .... , the convicted of assassin of civil rights ....
DOCm
S4: .... , the man convicted of killing ....
Sk: ....
Topic Modeling by LDA
Topic Distribution for each Document
DOC1
DOC2
DOCm
● Find the best topic for each document
● Get the most frequent best topic
Find topic words from most frequent best topic
assassin
shot
......

Figure 6.2 Finding Topic Words for the *Kill* Relation

Figure 6.2 gives a general flow of finding topic words for the *Kill* relation at one iteration. Steps for learning trigger words can be given as follows.

Step 1. Select *n* sentences for a relation type, *i*=0

Step 2. Random split sentences into *m* subsets ($m \leq n$)

Step 3. Aggregate *m* subset of sentences into *m* documents

Step 4. Train a topic model over *m* documents

Step 5. Find the best topic that has the highest probability for each document

Step 6. Select the most frequent best topic T_i among m documents

Step 7. Obtain the topic words under the most frequent best topic T_i

Step 8. *i*++, if $i < l$, go back to Step 2.

For each relation type, we first selection *n* sentences (step 1). From steps 2-7, we will generate a set of topic words for the target relation at each iteration *i*. Since the sentences are randomly split into *m* subsets, we might obtain a different set of topic words for different iterations. To obtain a robust set of topic words, we run steps 2-7 *l* times to obtain *l* sets of topical words. Then topical word frequency will be calculated from the *l* sets. Finally, the most frequent *K* topical words will be chosen for the target relation. Examples of trigger words for five relations are listed in Table 6.1,

Relation Type	**Trigger Words**
kill	assassin, shot, convicted, killing, …
org_founders	founded, founder, leader, president, …
people.education	degree, professor, graduate, …
people.sibling	brother, sister, younger, …
place_lived	living, moved, grew, residing, …

Table 6.1 Example Trigger Words

In Table 6.1, these trigger words are informative for their corresponding relations and can be used as semantic constraints in relation extraction.

As previously mentioned, another factor that can affect relation extraction is the entity type matching issue. For the *person.place_lived* relation, one entity must be a Person and the other entity must be a Location where the person lives. Take the following example:

> ***Tim/Bluhm*** *-LRB- born July/22/,/1970 -RRB- is a musician , producer , and songwriter , born in 1970 in California ; he lives with his wife , fellow musician Nicki/Bluhm , in* ***San/Francisco*** *.*

The person "*Tim/Bluhm*" lives in the location "*San/Francisco*". In this thesis, the Stanford CoreNLP tool is used to identify entity types. Then, we incorporate trigger words and entity types directly into our graphical model which will be discussed in Section 6.1.2 Profiling Model.

6.1.2 Profiling Model

The two semantic constraints *relation-specific trigger words* and *entity types* will be incorporated into the learning model for relation extraction. That is, we need to model (1) relations between two entities in a sentence and (2) the compatibility between entities, entity types and trigger words.

In this work, we choose to use the undirected graphical model (Lafferty et al., 2001) with the relation types as a hidden variable. A graphical model is actually a factor graph with a hidden variable y whose value is defined by $\boldsymbol{x}$ which represents a list of input features. In this thesis, y is the predicted relation type. A factor graph represents more explicitly the factorization of the underlying probability distribution among variable nodes. In a factor graph, factors are usually formulated as an exponential function of weighted features, that is, $\varphi(\boldsymbol{x}, y) = \exp(\boldsymbol{\theta} \cdot \overrightarrow{\mathbf{f}(\boldsymbol{x}, y)})$, where $\vec{\mathbf{f}}$ is a vector of feature functions and $\boldsymbol{\theta}$ is a vector of model parameters. To incorporate the two semantic constraints into the factor graph model, we treat the general bias of the model towards a particular relation for two entities as the factor φ_1 and the two additional semantic constraints and their related observations as the factor φ_2. The conditional probability distribution over these hidden and observation variables is defined as the product of φ_1 and φ_2, which is formulated as follows,

$$p(r_{ij}|\boldsymbol{x}) \propto \varphi_1(m_i, m_j, r_{ij})\varphi_2(m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}}, r_{ij})$$

where m_i and m_j are the i^{th} and the j^{th} entity mentions used by φ_1 and φ_2; t_i and t_j are the corresponding entity types used in φ_2, and $\boldsymbol{w}_{r_{ij}}$ are the trigger words for the relation r_{ij} in φ_2. The factor φ_1 assesses the general bias of the model towards a particular relation for two entities. Take the text below as an example:

> ***Robert/Rynasiewicz*** *is a professor of Philosophy at* ***Johns/Hopkins/University*** *and an Adjunct/Professor/in Philosophy …*

"*Robert Rynasiewicz*" and "*Johns Hopkins University*" should participate into the *employment_tenure* relation. It is defined over the relation variable r_{ij} and its corresponding entity mentions m_i and m_j, as formulated by,

$$\varphi_1(m_i, m_j, r_{ij}) = e^{\sum_k \lambda_k f_k(m_i, m_j, r_{ij})}$$

where λ_k is the weight for the k^{th} feature function $f_k(m_i, m_j, r_{ij})$. The following shows an example feature function,

$$f_k(m_i, m_j, r_{ij}) = \begin{cases} 1, if\ r_{ij} = employment\ and\ m_i = Robert\ Rynasiewicz \\ \qquad\qquad and\ m_j = Johns\ Hopkins\ Unviersity \\ 0, otherwise \end{cases}$$

This example feature returns 1 if r_{ij} is *employment_tenure* and there are two entity mentions: *"Robert Rynasiewicz"* and *"Johns Hopkins University"*; and 0 otherwise.

The factor φ_2 examines the compatibility between entity mentions, entity types and trigger words. It is defined by,

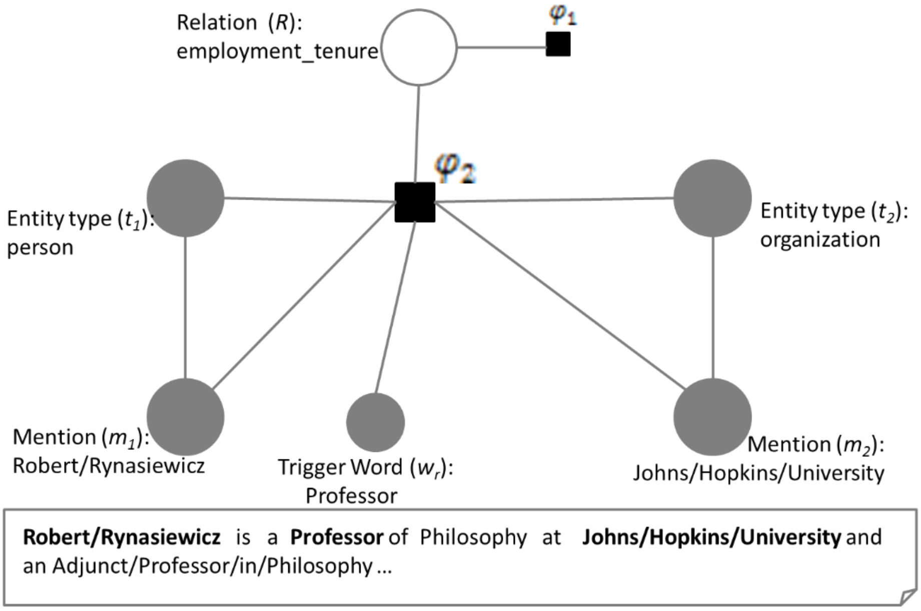
$$\varphi_2\left(m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}}, r_{ij}\right) = e^{\sum_l \mu_l g_l(m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}}, r_{ij})}$$

Similarly μ_l is the weight for the l^{th} feature function g_l. The following shows an example feature function,

$$g_l(m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}}, r_{ij}) = \begin{cases} 1, if\ r_{ij} = employment\ and\ m_i.Type = PER\ \ and \\ \qquad m_j.Type = ORG\ and\ tw = professor\ in\ \boldsymbol{w}_{r_{ij}} \\ 0, otherwise \end{cases}$$

This feature fires when the first entity is a *Person* and the second entity is an *Organization*. And the word *professor* is a trigger for the target relation.

Figure 6.3 shows some details of an instantiated factor graph for the *employment_tenure* relation between two entity mentions ("*Robert Rynasiewicz*" & "*Johns Hopkins University*"), their types (person and organization), and trigger word (professor). In general, the hidden variables encode various relationships among the entities: for example, r_{ij} indicates the most likely relation between two entity mentions m_i and m_j.



Robert/Rynasiewicz is a **Professor** of Philosophy at **Johns/Hopkins/University** and an Adjunct/Professor/in/Philosophy ...

Figure 6.3 Factor Graph for Measuring the Compatibility between Entities, Entity Types and Trigger Words.

6.1.3 Parameter Estimation

In this graphical model, the factors φ_1 and φ_2 compute the inner product between the vectors of features (f_k and g_l, sometimes called sufficient statistics) and parameters ($\{\lambda_k\}$ and $\{\mu_l\}$). In the two factor templates, higher positive weights (λ_k or μ_l) imply the corresponding features contribute more to the target relation whereas the negative weights will downgrade the contribution of the feature function for that relation.

Given these observation variables m_i, m_j, t_i, t_j and $\boldsymbol{w}_{r_{ij}}$, the inference procedure is to compute the marginal distribution $p(r_{ij}|m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}})$ or find the most likely relation assignment $\hat{r} = \arg\max_{r_{ij}} p(r_{ij}|m_i, m_j, t_i, t_j, \boldsymbol{w}_{r_{ij}})$. Maximum a posterior inference (MAP) is used to predict relations for a new pair of entities. In this task, Gibbs sampler is used. It randomly selects a relation variable

r_{ij} and samples the relation value conditioned on all the remaining observation variables. At test time, the temperature of the sampler is decreased to find an approximation of the MAP estimate, thus assigning larger weights to higher probable relation variable.

The goal of learning is to find a configuration to the parameters $\Lambda = \{\lambda_k,\ \mu_l\}$ that yields the highest prediction for $\hat{r}$. However, learning parameters in complex factor graphs is a very challenging task because computing gradients requires inferences over the full dataset before the parameters are updated. The SampleRank method remedies this problem by performing parameter updates within each step of the MCMC inference (Wick et al., 2009). It computes gradients between neighboring configurations in an MCMC chain. Parameters are updated when the model's ranking of any pair of neighboring configurations disagrees with the ranking by the objective function (the ground truth function). Suppose at the current time step t, we have the sample r_t and let the previous sample be r_{t-1} at time step $t-1$ in the chain, a perceptron-style update of model parameters is taken in case of disagreement of objective function and model ranking,

$$\Lambda^{\mathrm{t}} = \Lambda^{\mathrm{t-1}} + \eta(\emptyset(r_t, \boldsymbol{x}) - \emptyset(r_{t-1}, \boldsymbol{x}))$$

where $\emptyset: Y \times X \rightarrow \mathbb{R}^{|\Lambda|}$ refers to feature functions between relations and sequence of inputs. η is learning rate. In this work parameter estimation is done by running the SampleRank in Gibbs sampler using AdaGrad updates with Hamming loss (Duchi et al., 2011).

6.1.4 Features for PPE

We now describe the features we have used in this graphical model. For the factor φ_1, two features are used:

(1) **Surface text of entity mentions**, for example, the person *"Robert Rynasiewicz"*, we can have two surface features extracted: *Robert* and *Rynasiewicz*.

(2) **Part of speech of the entity mentions**, for the person "*Robert Rynasiewicz*", we can have parts of speeches of NNP for "*Robert*" and NNP for "*Rynasiewicz*" generated by the Stanford CoreNLP tool.

For the factor φ_2, we use the following features,

(1) **Entity type**: entity type feature, for example, "*London*" is a *Location*, "*Defense/Ministry*" is an *Organization*, and "*Carlos/Santana*" is a *Person*.

(2) **Trigger word**: we assign the corresponding relation to a trigger word, for example, we can have "*married*" linked to *person.marriage* and "*capital*" to *location.capital_of*, if they are in the sentences where the two entity mentions are located in.

(3) **Entity type + trigger word**: we combine the entity type with trigger words. If a sentence with the two entities has trigger word (s), we will associate the entity types with the trigger word (s), for example, LOCATION + born, PERSON + brothers and so on.

(4) **Entity mention + part of speech**: we link the entity mentions with their corresponding part of speeches, for example, Reagan + NNP and Normandy + NNP.

(5) **First entity type + token + second entity type**: this feature allows one token in-between two entities and only entity types are used in order to have a wide coverage of entity mentions. This feature is introduced because we observe that in terms of the *location.capital_of* relationship, entities are simply separated by the a comma, for example,

Marc/Laurick, born August/20/,/1963 in ***Trenton*** *,* ***New/Jersey*** *, is a*

Seattle-based bass player , songwriter , singer , and producer .

We can see that between “***Trenton***” and “***New Jersey***” is the comma symbol and we know that “***Trenton***” is the capital city of “***New Jersey***”. We have randomly sampled 500 sentences from the training data, found that 301 (60.2%) sentences are expressed in this way. For example, *Austin, Texas; Warsaw, Poland; Harbin, Heilongjiang* all have the pattern *LOCATION_,_LOCATION* and the first entities are capital cities of the second entities.

(6) **Context features**: we use the contextual features around the entity mentions, cases are:

Tokens or POS tags before the first entity

Tokens or POS tags after the second entity

Tokens or POS tag between the first and second entities

Examples for contextual features can be,

real, JJ; acquired, VBN

Attacks, NNS; Database, NNP; begin, VB

In addition to these features, we also use the regular expression patterns for the context tokens between two entities, before the first entity and after the second entity. They are given in Table 6.2.

Features	**Regular expressions**
All capitals	Token matches [A-Z] +
Numeric number	Token matches [0-9]+
Punctuation	Token matches [-,\\.;::?!()]+
Prefix	Length of token prefix is 3

Table 6.2 Regular Expression Features

All the features from (1) to (6) are categorical, which are converted into binary features.

(7) **Distance value**: we employ the distance value between two entities. Distance between entities is an integer. For these distance features, we apply a method to bin the features and convert them into categorical values.

6.2 Performance Evaluation

6.2.1 Experimental Setup

To evaluate the effectiveness of our semantic based approach, experiments are conducted on two datasets: (1) CoNLL-2004[12] with the sentences taken from the TREC corpus; (2) Wikipedia from which the Freebase is used to extract the relations. Precision (**P**), recall (**R**) and a harmonic F-score (**F**) are used for performance measures. Since the CoNLL-2004 dataset was created by Roth and Yih (2007), we can directly use it for evaluation. For the Wikipedia dataset, labeled training and testing data are extracted automatically using the facts from the Freebase. To extract trigger words, we split sentences into *m* subsets (*m*=20 for CoNLL-2004 and *m*=50 for Wikipedia) for each relation. Then we run a topic modeling procedure with hyper-parameters α=0.1 and β=0.1 by 100 iterations. The number of topics is heuristically set to 50 for both datasets. To generate a robust set of trigger words, we repeat the steps 20 times. In so doing, we obtain 10 semantic lexical words for each relation.

To predict relations over the two datasets, we rank the test examples by sampling 20 iterations with a low temperature of 0.0001 in the Gibbs sampler using the tool Factorie (McCallum et al., 2009). For convenience reasons, we use the

[12] http://cogcomp.cs.illinois.edu/Data/ER/conll04.corp

author name initials to name all the algorithms compared in our evaluations. Our algorithm is labeled as XL. To evaluate the effectiveness of the two semantic constraints, we use **ET** to denote the entity type features and **TW** to denote the trigger word features. We use plus (+) and negative (–) signs to indicate if such features are being used or not. Consequently, we have the following four variants of our XL algorithm:

(1) XL$_{\mathbf{-ET-TW}}$, a system that uses neither entity types nor trigger words, and is the baseline.

(2) XL$_{\mathbf{+ET-TW}}$, a system that uses entity types but has no trigger words.

(3) XL$_{\mathbf{-ET+TW}}$, a system that contains no entity types but uses trigger words.

(4) XL$_{\mathbf{+ET+TW}}$, a system that uses both entity types and trigger words.

6.2.2 Experiments on CoNLL-2004 Data

In the CoNLL-2004 dataset, five relations are given: *Located_In(Location*$_1$*, Location*$_2$*)* shows that *Location*$_1$ is in *Location*$_2$; *Work_For(Person, Organization)* expresses a *Person* works for an *Organization*; *OrgBased_In(Organization, Location)* indicates an *Organization* is based in a *Location*; *Live_In(Person, Location)* means a *Person* lives/resides in a *Location* and *Kill(Person*$_1$*, Person*$_2$*)* indicates a *Person*$_1$ killed *Person*$_2$. In case an input does not contain any predefined relation, we introduce an additional relation *NONE* indicating that no relation existing between entities. We extract these instances with *NONE* labels from sentences that contain none of the five relations. We then run 5-fold stratified cross-validations and make a comparison for each relation separately.

We then compare five relations with the four previous systems: RY07 Pipeline and RY07 Joint (RY are the initials of Roth and Yih) (2007), KM Card-pyramid and

KM Pipeline (KM are the initials of Kate and Mooney). The word *pipeline* refers to the process of handling entity extraction and relation extraction in two separate steps whereas *joint* refers to the combined method of both tasks into one joint step, making it easier to correct inter-related errors. The comparison results are listed in Table 6.3.

Systems	**P** (%)	**R** (%)	**F** (%)
RY07 Pipeline	64.6	54.88	57.24
RY07 Joint	68.46	54.02	58.14
KM Card-pyramid	73.04	62.66	*66.36*
KM Pipeline	75.08	60.2	66.28
XL$_{\mathbf{+ET+TW}}$	**76.46**	**83.31**	**79.49**

Table 6.3 Overall Comparison for the Five Relation Types

Table 6.3 shows that we achieved a 79.49% average F-score, an increase of 13.13% in F-score when compared to the KM Card-pyramid approach (66.36%). It is also worth noting that our system has a larger increase in recall and this increase comes from the usage of an entity type feature in the graphical model. The reason for the tiny increase in precision lies in the fact that the trigger words learned for the *Located_In*, *Work_For* relations are not closely linked to theses relations. This problem arises due to a filtering strategy by simply using Noun, Verb and Adjective in topic modeling procedure. We then evaluate the effectiveness of the two semantic constraints by the precision (**P**), recall (**R**) and F-score (**F**). The comparison result is plotted in Figure 6.4.

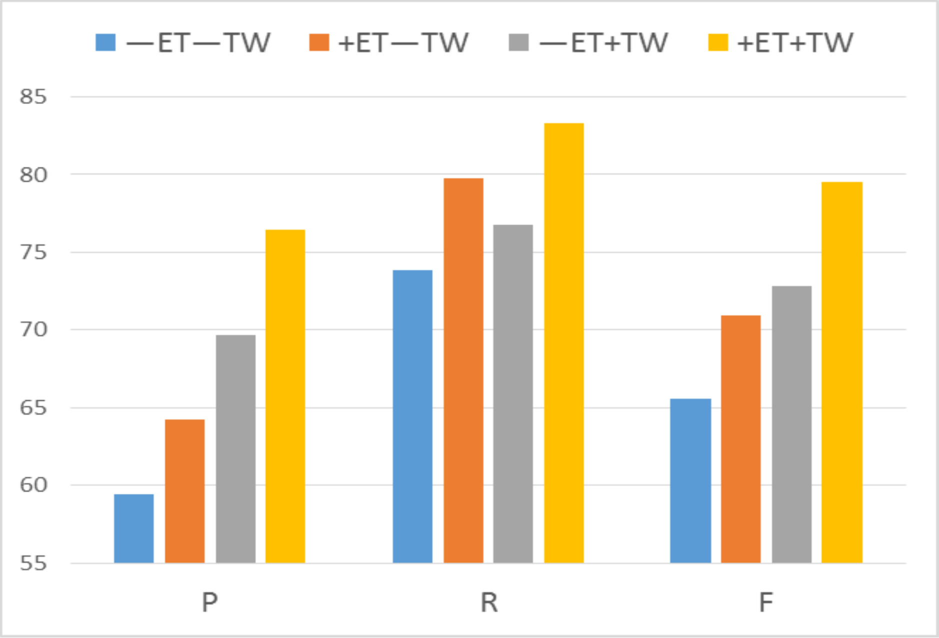


Figure 6.4 Comparison of Variants of Our Algorithm using CoNLL-2004

Figure 6.4 shows that a combination of entity type and trigger words will improve system performance in precision and recall for the five relations. Among the four variants of our XL algorithm, XL$_{-ET+TW}$ has a larger increase in precision due to the usage of trigger words, and entity type constraint contributes more to the recall increase in XL$_{+ET-TW}$. Following the comparison methods taken by Kate and Mooney (2010), we compare five relations separately with the four systems.

Systems	**Located_In**		
	P (%)	**R** (%)	**F** (%)
RY07 Pipeline	52.5	56.4	50.7
RY07 Joint	53.9	55.7	51.3
KM Card-pyramid	67.5	56.7	58.3
KM Pipeline	71.5	57	62.3
XL$_{+ET+TW}$	58.04	**73.64**	**64.68**

Table 6.4 Comparison for the *Located_In* Relation Type

Systems	**Work_For**		
	P (%)	**R** (%)	**F** (%)
RY07 Pipeline	60.8	44.4	51.2
RY07 Joint	72	42.3	53.1
KM Card-pyramid	73.5	68.3	70.7
KM Pipeline	**74.1**	66.0	69.7
XL$_{+ET+TW}$	71.1	**78.07**	**74.31**

Table 6.5 Comparison for the *Work_For* Relation Type

Systems	**OrgBased_In**		
	P (%)	**R** (%)	**F** (%)
RY07 Pipeline	77.8	42.1	54.3
RY07 Joint	**79.8**	41.6	54.3
KM Card-pyramid	66.2	64.1	64.7
KM Pipeline	70.6	60.2	64.6
XL$_{+ET+TW}$	79.74	**83.83**	**81.54**

Table 6.6 Comparison for the *OrgBased_In* Relation Type

Systems	**Live_In**		
	P (%)	**R** (%)	**F** (%)
RY07 Pipeline	58.9	50.0	53.5
RY07 Joint	59.1	49.0	53.0
KM Card-pyramid	66.4	60.1	62.9
KM Pipeline	68.1	56.6	61.7
XL$_{+ET+TW}$	**81.7**	**88.1**	**84.75**

Table 6.7 Comparison for the *Live_In* Relation Type

Systems	**Kill**		
	P (%)	**R** (%)	**F** (%)
RY07 Pipeline	73.0	81.5	76.5
RY07 Joint	77.5	81.5	79.0
KM Card-pyramid	91.6	64.1	75.2
KM Pipeline	91.1	61.2	73.1
XL$_{+ET+TW}$	**91.74**	**92.91**	**92.17**

Table 6.8 Comparison for the *Kill* Relation Type

Experimental results from Table 6.4 to Table 6.8 show that our system

XL_{+ET+TW} has remarkably outperformed other systems in F-scores. Experimental results show that our XL algorithm gains a significant increase in precision and recall for the *Live_In* and *Kill* relations. This is attributed to the set of automatically learned trigger words that are closely tied to the target relation. Take the *Kill* relation for example, the trigger words are: *assassination*, *death*, *killing* etc. For the *Live_In* relation, we have trigger words like *native, home*, *born* etc. But, for the *Located_In* relation, the trigger words such as *officials*, *government*, *state* etc. cannot explicitly demonstrate their connection with the target relation type. The increases in F-scores are credited to the entity type constraints. This is the same for the *Work_For* and *OrgBased_In* relations. The 10 trigger words for the five relation types are listed in Table 6.9.

Relation Types	**Trigger Words**
Located_In	year, officials, government, state, president, people, today, fire, reported, report
Live_In	today, born, native, president, year, home, years, text, state, government
OrgBased_In	report, daily, text, economic, president, people, year, type, country, state
Work_For	president, government, told, year, people, years, officials, spokesman, state, report
Kill	assassination, shot, convicted, killing, assassin, years, death, killed, assassinated, fired

Table 6.9 Trigger Words for the CoNLL-2004 Dataset

Table 6.9 shows that the *Kill* and *Live_In* relations have a well learned set of trigger words. The trigger words for the other three relations cannot explicitly show their close connection to the target relations. This problem occurs because most of the time these relations are not clearly expressed by trigger words (Noun, Verb and Adjective). Take the *Located_In* relation for example,

Officials in ***Perugia*** *in* ***Umbria*** *province said five people were arrested there*

Tuesday night after police stopped their car and found $1/million in bogus bills in the trunk.

Right now, the fire season is just starting to gear up, said Sandi/Sacher, spokeswoman at the federal government's wildfire command post in ***Boise** , **Idaho**.*

Example sentences show that *Located_In*(***Perugia**, **Umbria***) and *Located_In*(***Boise** , **Idaho***) relations are not explicitly expressed with either a preposition *in* or a punctuation symbol comma. In comparison, for the *Kill* relation, trigger words are often used, for example,

In 1969, ***James/Earl/Ray*** *pleaded guilty in Memphis/,/Tenn., to the assassination of civil rights leader* ***Martin/Luther/King/Junior****.*

William/Leonard/Jennings *sobbed loudly as he appeared in a magistrates court in West/Yorkshire and was charged with killing his 3-year-old son,* ***Stephen****, who was last seen alive on Dec. 12, 1962.*

The two example sentences give the relation pairs of *Kill*(***James/Earl/Ray***, ***Martin/Luther/King/Junior***) and *Kill*(***William/Leonard/Jennings***, ***Stephen***). Clearly, the *Kill* relation types are triggered by either *assassination* or *killing*. Therefore, topic models can find the trigger words that are closely related to the *Kill* relation type. These trigger words boost a larger increase in both precision and recall as can be shown in Table 6.8.

After discovering this problem, we decide to manually prepare a set of trigger words without referring to the training and testing dataset and test the effectiveness of our approach in trigger word generation. The manually prepared trigger words are given in Table 6.10.

Relation Types	Trigger Words
Located_In	COMMA, in , at
Live_In	lived, lives, living, reside, resided, resides, residing, resident, residence
OrgBased_In	based, headquarter, headquartered, headquarters
Work_For	professor, official, heads, placekicker, spokesman, secretary-general, secretary, peacekeepers, capt., captain, chairman, republican, director, executive, leader, president, actors, solicitor, judge, publisher, shearer, head, electrician, manager, employee, spokeswoman, columnist, attorney, economist, analyst, engineer, physiologist, entomologist, ecologist, rep., representative, correspondent, inspector, reporter, commissioner, director-general, pilot, anchorman, lobbyist, anthropoligist, sen., quaestor, commander, editor, activist, designer, minister, superintendent, secretaries
Kill	kill, assassination, assassinated, assassinating, slaying, killing, assassin, murdering, murder, murdered, killed, killer, shooting, slayings, murderers

Table 6.10 Manually Prepared Trigger Words for the CoNLL-2004 Dataset

We made a comparison using LDA-generated and manually prepared trigger words. Comparison results are plotted in Figure 6.5.

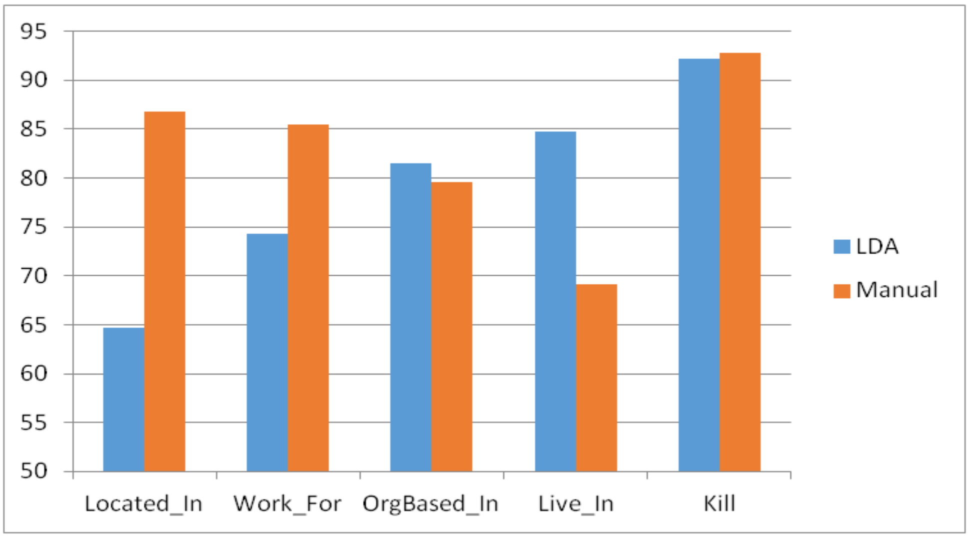


Figure 6.5 Comparison of the Five Relations using LDA-generated and Manually Prepared Trigger Words

Figure 6.5 clearly shows that manually prepared trigger words are far better than the LDA-generated ones in the *Located_In* relation. This implies that the *Located_In* relation in the CoNLL-2004 dataset is represented by the pattern that uses either a *comma* or prepositions *in* and *at*. In terms of *Work_For* relation, the number of manually prepared trigger words is greater than 10. This would require heavier annotation efforts and makes relation extraction a tedious task. Interestingly, the LDA-generated trigger words give a better performance than the manually annotated ones in the *Live_In* relation. This is because "born", "native" and "home" can represent the *Live_In* relation better than "lived" and "reside" on the CoNLL-2004 dataset. In addition, for the *Kill* relation, the LDA-generated triggers are quite similar to the manually prepared ones, reaching an almost equal performance in F-score. Through the comparison between the LDA-generated and manually prepared triggers, we found that manually generated trigger words can perform well if they capture the pattern in a specific domain, otherwise they would perform poorly.

6.2.3 Experiments on Wikipedia Data

Supervised methods for relation extraction need human tagged training data, which is costly and time intensive. To automatically generate training data, we take a distantly supervised approach for relation extraction. Labeled training and testing data are extracted automatically using the facts from the Freebase. Freebase is a community-curated database that contains facts about named entities and their relations. We used the Freebase snapshot of June 23, 2013 in RDF format. In total, this snapshot provides 27,538 relations. We extracted the 39 most frequent relations from the Freebase, and map them to the 18 predefined relation types. The mapping is needed because some of the relation types in Freebase are equivalent. Others might simply be variations of the same relation due to rotation of the entities.

Relation Types	**Freebase Relations**
person.birthplace	people.person.place_of_birth
	location.location.people_born_here
person.deathplace	people.deceased_person.place_of_death
person.sibling	people.person.sibling_s
	people.sibling_relationship.sibling
	people.person.sibling_s..people.sibling_relationship.sibling
person.place_lived	people.person.places_lived..people.place_lived.location
person.nationality	people.person.nationality
person.marriage	people.marriage.spouse
	people.person.spouse_s..people.marriage.type_of_union
	people.marriage_union_type.unions_of_this_type..people.marriage.spouse
	people.person.spouse_s..people.marriage.spouse
	people.person.spouse_s
person.parents_children	people.person.parents
	people.person.children
person.education	people.person.education..education.education.institution
	people.person.education
organization.founders	organization.organization_founder.organizations_founded
	organization.organization.founders
organization.parent_child	organization.organization.parent
	organization.organization_relationship.parent
	organization.organization.parent..organization.organization_relationship.parent
	organization.organization_relationship.child
	organization.organization.child..organization.organization_relationship.child
organization.headquarters	organization.organization.headquarters
location.capital_of	location.capital_of_administrative_division.capital_of..location.administrative_division_capital_relationship.administrative_division
	location.administrative_division.capital..location.administrative_division_capital_relationship.capital
country.administrative_divisions	location.administrative_division.country
	location.country.administrative_divisions
business.employment_tenure	business.employer.employees..business.employment_tenure.person

	people.person.employment_history..business.employment_tenure.company
location.neighborhood	location.neighborhood.neighborhood_of
	location.place_with_neighborhoods.neighborhoods
location.contains	location.location.containedby
	location.location.contains
location.partially_contains	location.location.partially_contains
	location.location.partially_containedby
organization.membership	organization.membership_organization.members..organization.organization_membership.member
	organization.organization_member.member_of..organization.organization_membership.organization

Table 6.11 Freebase Relations Mapping

In this work, we use the Wikipedia dump with the timestamp: April 03, 2013. A Wikipedia extractor tool is used to extract 4,064,234 articles from this dump and the Stanford CoreNLP tool is used to preprocess these articles, including tokenization, part-of-speech tagging, and named entity recognition. For named entities, we treat consecutive tokens with the same category as a single entity mention. Then we associate these mentions with Freebase entities by simple string matching and extract sentences containing entity mentions sharing a relation in the Freebase. Then, the Wikipedia articles are split into two separate training and testing sets with a ratio at about 2.33. Statistics of the splits are shown in Table 6.12. The instances refer to the number of sentences used for the relation extraction in our algorithm.

	Train	Test
#Documents	2,844,964	1,219,270
#Instances	735,615	316,675

Table 6.12 Statistics of Wikipedia

In this set of experiments, we simply use the 18 relation types without the *NONE* relation type. This is because some pairs of entity mentions that can be labeled as *NONE* may actually have some kind of relation due to the lack of Freebase coverage. Table 6.13 shows the average performance of our four variants of our XL algorithms across the 18 relation types.

Systems	**P** (%)	**R** (%)	**F** (%)
XL$_{\mathbf{-ET-TW}}$	84.95	80.64	82.32
XL$_{\mathbf{+ET-TW}}$	87.69	84.34	85.69
XL$_{\mathbf{-ET+TW}}$	91.88	89.08	90.31
XL$_{\mathbf{+ET+TW}}$	**95.82**	**94.29**	**95.00**

Table 6.13 Comparison of Four Variants of Our XL Algorithm in Wikipedia

Table 6.13 shows that when compared to the baseline XL$_{\mathbf{-ET-TW}}$, entity type feature XL$_{\mathbf{+ET-TW}}$ helps to increase precision and recall by 2.74% and 3.7%, respectively; trigger words XL$_{\mathbf{-ET+TW}}$ has 6.93% increase in precision and 8.44% increase in recall. Comparatively speaking, trigger words are a more effective single feature compared to entity type as trigger words alone give the best improvement for both precision and recall. This is because trigger words are directly associated with relation types. The larger increase in precision and recall by XL$_{\mathbf{-ET+TW}}$ is attributed to the fact that trigger words are accurately extracted from the large number of sentences in Wikipedia. For example, "*headquartered*", "*based*", "*headquarters*" are extracted for the *organization.headquarters* relation; "*son*", "*daughter*", "*father*" are detected for the *person.parents_children* relation; "*founded*", "*founder*" are learned for the *organization.founders* relation; and so on.

Table 6.13 also shows that entity type constraint increases the recall of XL$_{\mathbf{+ET+TW}}$ remarkably by 4.69% when compared to XL$_{\mathbf{-ET+TW}}$. After integrating both entity types and trigger words, our system XL$_{\mathbf{+ET+TW}}$ obtains 10.87% and 13.65% increments in precision and recall when compared to the baseline XL$_{\mathbf{-ET-TW}}$. We then check the top 10 trigger words in Table 6.14 for the 18 relation types.

Relation Types	**Trigger Words**
person.birthplace	born, player, professional, football, plays, footballer, retired, singer, actress, actor
person.deathplace	died, born, politician, painter, death, killed, bishop, composer, executed, actor
person.sibling	born, film, older, brother, brothers, younger, actor, sister, son, sisters
person.place_lived	living, born, moved, grew, artist, lives, lived, residing, works, musician
person.nationality	born, player, footballer, professional, football, politician, film, singer, minister, president
person.marriage	film, wife, son, starring, born, written, actress, directed, daughter, husband
person.parents_children	born, son, daughter, wife, father, brother, mother, actress, actor, film
person.education	born, professor, attended, degree, received, graduated, graduate, educated, football, studied
organization.founders	born, founded, film, founder, company, label, president, organization, leader, produced
organization.parent_child	division, owned, subsidiary, company, part, operated, channel, developed, resolution, game
organization.headquarters	company, headquartered, based, owned, bank, largest, headquarters, video, game, developer
location.capital_of	located, born, km, municipality, province, state, commune, region, city, district
country.administrative_divisions	administrative, located, municipality, district, town, village, central, north, born, commune
business.employment_tenure	born, director, professor, president, history, chairman, ceo, executive, science, chief
location.neighborhood	located, neighborhood, area, city, district, street, school, building, borough, historic
location.contains	located, born, town, village, municipality, district, city, state, historic, administrative
location.partially_contains	located, river, tributary, western, central, eastern, state, region, north, bridge
organization.membership	member, permanent, born, ambassador, representative, general, secretary, government, council, resolution

Table 6.14 Trigger Words for the Wikipedia Dataset

Table 6.14 shows a close connection between trigger words and their relation types, for example, we have the lexical word *died* linked to *person.deathplace* relation, *living* to the *person.place_lived* relation, *headquartered* to the *organization.headquarters* relation, and so on. In experiments over this dataset, we also manually compile a set of trigger words for the Wikipedia 14 relations. Details of these relations are listed in Appendix 2. There are four common relation types which are difficult to identify trigger words as listed in Table 6.15. For these 4 types, we did give not any trigger words and did not compare these 4 types with the automatic extraction methods.

location.neighborhood	location.contains
organization.membership	location.partially_contains

Table 6.15 Relation Types without Manually Prepared Trigger Words

Hence, we experiment over the 14 relation types and make a comparison between the LDA-generated and manually compiled trigger words.

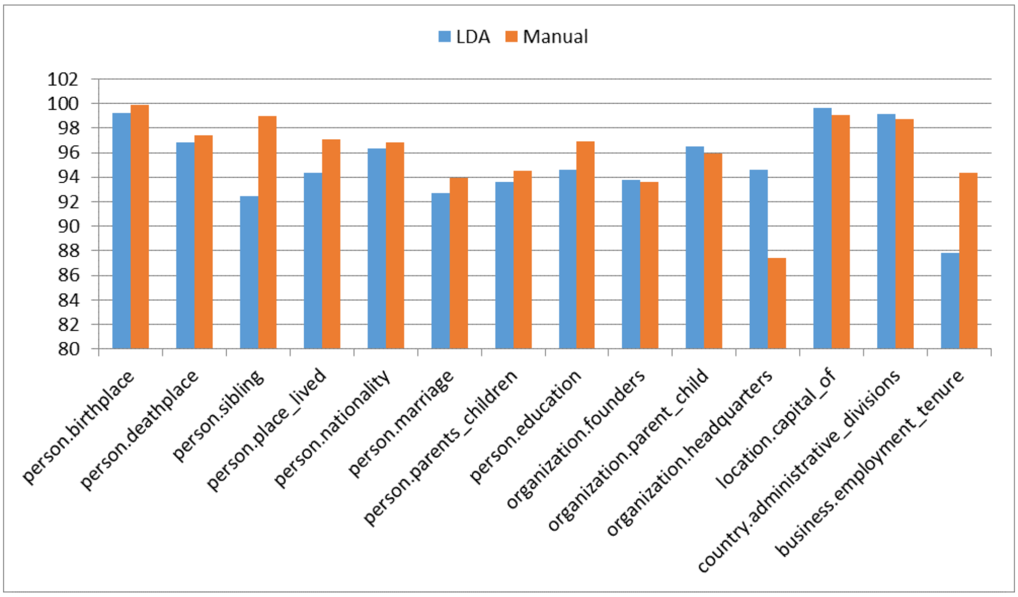


Figure 6.6 Comparison of the Fourteen Relations using LDA-generated and Manually Prepared Trigger Words on Wikipedia Data

Figure 6.6 shows that with the exception of the *person.sibling* and *business.employment_tenure* relations, the LDA-generated trigger words demonstrate a similar performance to the manually compiled ones. On the *organization.headquarters* relation, the LDA approach behaves far better than the manually compiled one. There are two reasons for the lower performances in the *person.sibling* and *business.employment_tenure* relation types. One is that the word "*born*" is wrongly assigned to the *person.sibling* and *business.employment_tenure* relations. After removing this word, we have the F-scores of *person.sibling* and *business.employment_tenure* relations increased from 87.82% to 88.25%, and 92.46% to 93.56%. The other reason is that the number of the manually compiled triggers is larger than that of the LDA-generated ones for each relation. On average, there are 26 lexical words for each relation.

6.2.4 Discussion

Experiments on the two different datasets all show that a combination of entity type and trigger word in the graphical model can significantly increase performance of relation extraction. In these experiments, we found that trigger words make a great contribution to system improvement. Relation-specific trigger words can greatly help to reduce false positives in relation extraction. And our approach has outperformed both the pipeline and joint approaches. This is because associating the trigger word with its corresponding relation type can provide much improved labeling of instances and entity type constraints can also effectively control the selection of the two arguments that participate in a relation. Similar to the slot filling task in TAC-KBP (Dang and Owczarzak, 2009; Ji et al., 2010; Ji et al., 2011), our relation extraction approach can also be applied to learning missed attributes from Web texts for a person in the Freebase, thus consolidating the structured knowledge about entities in the Freebase.

In principle, when our algorithm returns the *NONE* relation between two entities using Wikipedia, it does not necessarily mean the two entities have no relation due to the lack of coverage of Freebase. In the following three example sentences in Wikipedia:

> ***Wierzyca/Pelpin*** *is a football club based in* ***Pelpin*** *-LRB- Poland -RRB- .*
>
> *He also had a daughter* ***Chryse*** *, who married* ***Dardanus*** *and brought the Palladium to Troy .*
>
> ***Grayven*** *is the third son of* ***Darkseid*** *, born of an unknown mother , younger brother to Kalibak and Orion .*

The three *organization.headquarters, person.marriage, person.parents_children* relations do hold. But the Freebase contains no record for these relation instances. This is why we did not consider the *NONE* relation for the experiments in the Wikipedia dataset.

Another problem is that some relations are not directly expressed. Take the *person.marriage* relation for example in the following text,

> *Zelda/Rae/Williams , born July/31/,/1989 , is an American actress and the daughter of actor and comedian* ***Robin/Williams*** *and* ***Marsha/Garces*** *.*

"*Robin/Williams*" and *"Marsha/Garces*" are a couple, and they should participate in a *person.marriage* relation, but this kind of relation cannot be identified by trigger words nor relation types. The information is inferred based on the third person, their daughter "*Zelda/Rae/Williams*".

Additionally, we found that one sentence can contain more than one relation type. But our algorithm can only select one relation type. For example, in the following text:

> *It is known for the involvement of* ***Gulfstream/Aerospace*** *founder* ***Allen/Paulson*** *,*

who was CEO from 1994/to/2000 , and former Chrysler chairman Lee/Iacocca , who has been a major investor in the company since 1995 .

This sentence actually shows the *organization.founders* relation because "*Allen/Paulson*" is the founder of the "*Gulfstream/Aerospace"* organization; it also expresses the *business.employment_tenure* relation because of the trigger word "*CEO"* after the second argument "*Allen/Paulson"*. In our system, however, we only take the top ranked entity pair ("*Gulfstream/Aerospace"* and "*Allen/Paulson"*).

6.3 Chapter Summary

In this chapter, we present a novel approach to improve relation extraction using semantic constraints including both entity type and trigger words in a graphical model. Entity types control the selectional preference of arguments that participate in a relation. Trigger words add more positive evidences that are closely related to the target relations, which help reduce the use of noisy data. In the evaluation of the CoNLL-2004 dataset, we obtained a 79.49% average F-score, an increase of 13.13% compared to the state-of-the-art system. On the evaluation of Wikipedia data, we obtained a 95% average F-score, an increase of 12.68% compared to the baseline system without semantic constraints.

A major advantage of our approach is its extensibility because trigger words from any domain can be learned automatically and added into the graphical model for various relation types. This is particularly important for the distantly supervised relation extraction which extracts relations from a large knowledge base. On the other hand, our approach is currently restricted to only the three major entity types. We envision that it can be extended to other entity types, for example Date. This would be helpful in extracting the date/time for a person’s activity (birthdate, death date or date of marriage).

In this work, we choose to use Freebase as the knowledge base for the relation identification because it is publicly available. Also, the importance is to identify the relations in Freebase so we can go to Wikipedia to find training data. In principle, the methods developed in this chapter is not restricted to use Freebase only. It can also make use of other knowledge bases, for example, the Google Knowledge Graph and Microsoft Bing Satori (Singhal, 2012; Farber, 2013)[13] as long as we can obtain sentence based training data. Even though there are different reference sites where we can obtain training sentences, by our assessment, Wikipedia is still the best training data resources because of its wide coverage.

Chapter 7 Named Entity Linking

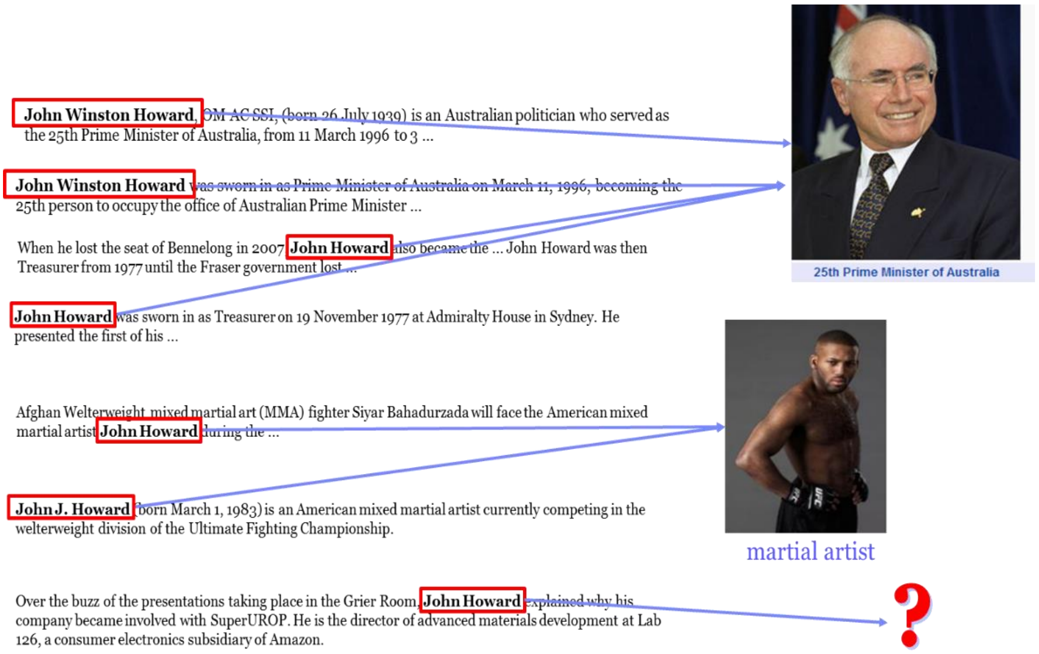
Optionally, to facilitate knowledge population, disambiguated persons with profiles can be linked to named entities in an existing knowledge base, such as Wikipedia. This process is called named entity linking (NEL in short). However, entity ambiguity (e.g. the mention “*John Howard*” can refer to the prime minister or the martial artist) is quite challenging for NEL. To solve this problem, both ranking methods (Han and Zhao, 2009; Varma et al., 2010; Xu et al., 2011) and classification approaches are proposed (Agirre et al., 2009; Varma et al., 2009). However, these NEL methods face the imbalanced data problem because the target entity in Wikipedia usually has only one corresponding article for a name mention. To solve

[13] These two knowledge bases cannot be used for this thesis work because we have no accessibility to these resources during the period of this work.

this problem, we attempt to enrich the target entity feature representation by using the outgoing pages embedded in the Wikipedia article, and then link mentions to the target Wikipedia entities using the SVM classifier.

7.1 Algorithm Design for NEL

Given a query name mention, NEL is to find the most likely knowledge base entity from a list of candidate entities. Take the name mention "*John Howard*" as an example,



John Winston Howard, OM AC SSI, (born 26 July 1939) is an Australian politician who served as the 25th Prime Minister of Australia, from 11 March 1996 to 3 ...

John Winston Howard was sworn in as Prime Minister of Australia on March 11, 1996, becoming the 25th person to occupy the office of Australian Prime Minister ...

When he lost the seat of Bennelong in 2007 **John Howard** also became the ... John Howard was then Treasurer from 1977 until the Fraser government lost ...

John Howard was sworn in as Treasurer on 19 November 1977 at Admiralty House in Sydney. He presented the first of his ...

Afghan Welterweight mixed martial art (MMA) fighter Siyar Bahadurzada will face the American mixed martial artist **John Howard** during the ...

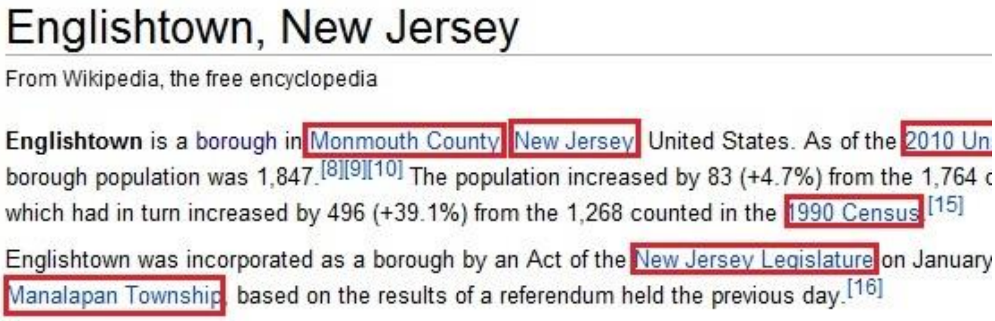
John J. Howard (born March 1, 1983) is an American mixed martial artist currently competing in the welterweight division of the Ultimate Fighting Championship.

Over the buzz of the presentations taking place in the Grier Room, **John Howard** explained why his company became involved with SuperUROP. He is the director of advanced materials development at Lab 126, a consumer electronics subsidiary of Amazon.

Figure 7.1 Linking Name Mentions for "*John Howard*"

to the Entities in the Wikipedia

In Figure 7.1, we tend to link the first four name mentions of "*John Howard*" to the prime minister and the two mentions in the middle to the martial artist. But the last mention has no corresponding entity in Wikipedia and NULL results will be returned. The challenge in NEL is that each entity in knowledge base (Wikipedia) has a single article. This results in insufficient evidences for a candidate entity to which the name mention can be linked. To tackle this problem, we observe that outgoing links embedded in a candidate page should be related to the candidate entity. Take the candidate entity "*Englishtown, New Jersey*" for example, its Wikipedia article is given as follows,



Englishtown, New Jersey

From Wikipedia, the free encyclopedia

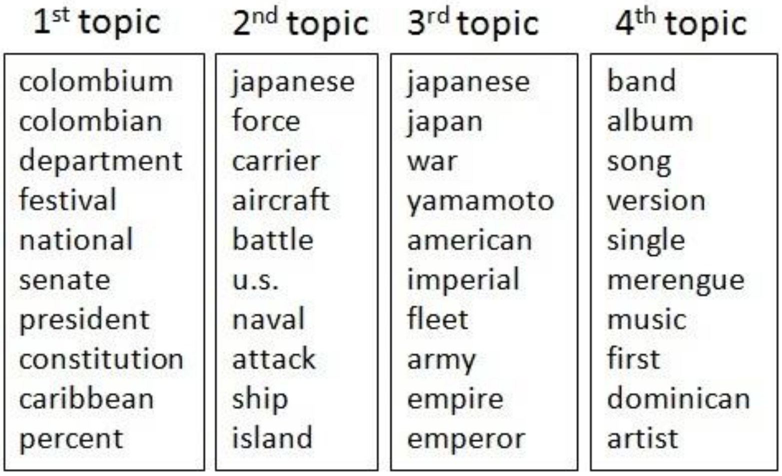
Englishtown is a borough in Monmouth County, New Jersey, United States. As of the 2010 Un
borough population was 1,847.[8][9][10] The population increased by 83 (+4.7%) from the 1,764 c
which had in turn increased by 496 (+39.1%) from the 1,268 counted in the 1990 Census.[15]

Englishtown was incorporated as a borough by an Act of the New Jersey Legislature on January
Manalapan Township, based on the results of a referendum held the previous day.[16]

Figure 7.2 Outgoing Links for "*Englishtown, New Jersey*"

In Figure 7.2, we can extract outgoing pages from the anchor texts in the red boxes, for example, "*Monmouth County, New Jersey*", "*1990 Census*", "*New Jersey Legislature*", and "*Manalapan Township*". These outgoing pages are treated equally as the candidate entity. In so doing, each candidate has a list of outgoing pages sharing the same class label.

Relying on outgoing pages, the next problem is to select features to represent the name mentions and the target candidate entities. We assume that documents that are close to each other are expected to have similar topic words. Take the mention "*Santa Cruz*" for example, after sampling topics from the mention document and candidate documents, we obtained lists of topic words under each topic in Figure 7.3.



1st topic	2nd topic	3rd topic	4th topic
colombium	japanese	japanese	band
colombian	force	japan	album
department	carrier	war	song
festival	aircraft	yamamoto	version
national	battle	american	single
senate	u.s.	imperial	merengue
president	naval	fleet	music
constitution	attack	army	first
caribbean	ship	empire	dominican
percent	island	emperor	artist

Figure 7.3 Four Topics for the Mention "*Santa Cruz*"

It is obvious that the first topic is about "politics", the second and third are about "war" and the fourth is related to the "music" topic. We then use these topic words as features and compute similarities between mentions and candidate entities. In this work, we investigated two strategies for selecting the most likely candidate for each name mention:

(1) Candidate selection using multi-class SVM

To select the most likely candidate for a query, we apply the multi-class SVM approach. Traditionally, the multi-class classification problem can be decomposed into binary classification tasks. The commonly used strategies include One-versus-all (OVA) and One-versus-one (OVO) (Aly, 2005; Milgram et al., 2006). The OVA strategy solves the multi-class problem by building one SVM for each class, which is trained to discriminate the samples in a given class from the samples in all of the other classes. When classifying a new instance, the classifier with the maximum output will be chosen and the corresponding class label will be given to the new instance. The OVO strategy builds one SVM for each pair of classes. If we have M classes, we have to build $\frac{M\times(M-1)}{2}$ binary classifiers. When classifying a new instance, a voting technique is employed to select the class with the maximum votes. In the named entity linking task, we have used the OVO strategy for selecting the most likely candidate.

(2) Candidate selection by maximizing similarity

In this approach, mentions and candidates are compared using the topic words with TFIDF scores. The candidate having the maximum similarity with name mentions is chosen as the most likely target entity. The maximum similarity is defined by,

$$\max_{c_i} cosine\ (m, c_i)$$

where c_i is i^{th} candidate for a name mention.

7.2 Performance Evaluation

We evaluate our approach on the dataset of TAC-KBP 2012 with a large corpus of newswire and web documents and a reference knowledge base (Ellis et al., 2012). The goal of NEL is to link mentions from the texts into the reference knowledge base (Wikipedia). It needs to generate candidates first and then select from the candidate list the most likely candidate entry for the name mention. In this thesis, we generate candidates using the methods in the work done by Xu et al. (2012). For the NULL results which have no corresponding Wikipedia entities, a simple Hierarchical Agglomerative Clustering algorithm is used to group them into clusters.

Based on the candidate articles and mention document, we run the topic modeling procedure to obtain the topic words for mentions and their candidate articles (Blei et al., 2003). The values for the hyper-parameters α and β are $\frac{50}{K}$ (K is the number of topics) and 0.01 (Steyvers and Griffiths 2006) and the number of iterations is set to 500. Note that the name mention documents and candidate documents are placed together to obtain the topic words for each document. 50 topic words under each topic are used. In terms of classifier, the LIBSVM[14] tool with one-versus-one strategy is used and default parameters are kept.

For evaluation, micro-average and BCubed$^+$ metrics are used. In terms of BCubed$^+$ metric, *L(e)* and *C(e)* denote the category and cluster for an item *e* respectively. *SI(e)* and *GI(e)* refer to the system and gold-standard knowledge identifier for the item *e*. the correctness of the relation between items *e* and *e'* is,

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

$$PrecisionBCubed^{+} = Avg_e \left[Avg_{e',C(e)=C(e')} G(e,e') \right]$$

$$RecallBCubed^{+} = Avg_e \left[Avg_{e',L(e)=L(e')} G(e,e') \right]$$

Finally, the harmonic mean of BCubed$^+$ precision and recall is used to rank the overall performance. The experimental results are given in Table 7.1.

Systems	**Micro Average**	**BCubed$^+$ Precision**	**BCubed$^+$ Recall**	**BCubed$^+$ F1**
SVM	0.129	0.093	0.121	0.106
Max Cosine	0.306	0.261	0.289	0.274

Table 7.1 Performance Evaluation of the Named Entity Linking

Given that the highest F1 score among all participants is 0.73 and the median F1 score is 0.536, the system performance is far below the median level. In the analysis, we found that our system has poor answer coverage and only has 41.15% coverage of the answers. Worse still, the NULL detection system has a poor performance since it has only generated 76 NULL results while the manual answer has 1049 NULL results. Additionally, using the outgoing pages in the candidate list

[14] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

would introduce noise as well, since the outgoing pages in the candidate documents may share no topical words with the candidates, thus bringing noises into the topic modeling process. On the other hand, the number of topics is crucial to the topic modeling. In this task, we simply set the number of topics to the size of candidates. If the number of topics is known beforehand, the latent topics can be modeled out of the document collections by topic modeling technique. In fact, the number of topics is expected to be estimated using such approaches as cross validation, the nonparametric Bayesian method of Hierarchical Dirichlet processes (Teh et al., 2006).

7.3 Summary for NEL

For the named entity linking, we enrich the candidate entities using the outgoing pages and rank candidates based on topic words sampled from mention document and candidate documents. The poor answer coverage and the detection of NULL results bring a great loss in F1 measure. In future, investigations will be conducted on finding suitable approaches to increase answer coverage and to handle the NULL detection problem. Furthermore, it is also possible to apply network similarity measures to select quality outgoing pages for the target candidate entity, for example, the label propagation algorithm (Zhu and Ghahramani, 2002).

Chapter 8 Conclusions and Future Works

This thesis presents a series of systematic studies on named entity disambiguation from web text. The studies begin with extraction of the salient keyword features for Web Person Disambiguation (WPD); then it proceeds to utilize the hierarchical co-reference technique to disambiguate person names; successively build personal profiles by incorporating two semantic constraints of trigger words and entity types into a graphical model. In case of personal description available in reliable resources such as Wikipedia, entities are linked directly to these resources. The main contributions of this work can be summarized as follows:

(1) Extraction of WPD semantic features using naturally annotated resources. We extract keywords as WPD features by training the CRFs model on a large keyword annotated corpus which is automatically generated from Wikipedia. In so doing, our keyword extractor requires no manual annotation efforts and is much less domain sensitive due to Wikipedia's wide coverage. Experiments are conducted on three publicly available datasets, and we have achieved the state-of-the-art performance and close to the state-of-the-art performance over all the tested datasets.

(2) Resolving name ambiguities by using co-reference technique. We disambiguate person names by applying the hierarchical co-reference model, which does not need to manually tune the number of clusters. This model can incorporate many person-specific features, for example, age, gender, keywords and so on. Experiments conducted on the WPD dataset show that the proposed method outperforms all systems that learn the threshold automatically. We also achieve a comparable performance with the top two systems which manually tune the number of clusters.

(3) Extraction of personal profiling using two constraints of trigger words and entity types. Trigger words are automatically learned for target relations and remarkably improve relation extraction performance. Entity types control the selectional preferences of two entities in a relation. Experiments conducted on the public dataset outperforms the state-of-the-art system remarkably. We also experiment on Wikipedia data using Freebase as a source for extracting relation facts, obtaining a larger increase in F-score compared to the baseline system which does not use semantic constraints.

In general, our methods are proven to be effective in named entity disambiguation. However, there are still issues that can be considered in future. Firstly, the features used in WPD and PPE are separately handled. That is, the two modules do not work in a bi-directional mode. However, since the features used for profile extraction should also be useful for person name disambiguation, a combined approached can be explored in the future. Secondly, for relation extraction, methods for cross-sentence relation extraction and indirect relation extraction can be explored, for example, feeding entities from different sentences into relation extraction model and studying the conjunction rules in our probabilistic graphical model. Thirdly, in named entity linking, methods for selecting quality outgoing pages can be investigated, for example, the label propagation algorithm can be used to select the outgoing pages that are closely related to the target candidate entity.

Our WPD algorithm works effectively on Web documents, but it is difficult to be used directly in social media data because social media text is much shorter to provide enough information. In social media environment, personal profile information and links among entities may play more important role for entity disambiguation and can be studied further.

Bibliography

Agichtein, E., Gravano, L. 2000. Snowball: Extracting Relations from Large Plain-text Collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85-94, San Antonio, Texas, USA.

Agirre, E., Chang, A. X., Jurafsky, D. S., et al. 2009. Stanford-UBC at TAC-KBP. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA.

Aly, M. 2005. Survey on Multi-Class Classification Methods. *Technical Report,* Caltech, *USA.*

Artiles, J., Gonzalo, J., Verdejo, F. 2005. A Testbed for People Searching Strategies in the WWW. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 569-570, Salvador, Brazil.

Artiles, J., Gonzalo, J., and Sekine, S., 2007. The SemEval-2007 WePS Evaluation: Establishing a Benchmark for the Web People Search Task. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007),* pages 64-69, Prague, Czech Republic.

Artiles, J., Gonzalo, J., Sekine, S. 2009. WePS 2 Evaluation Campaign: Overview of the Web People Search Clustering Task. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Artiles, J., Borthwick, A., Gonzalo, J., Sekine, S., Amigó, E. 2010. WePS-3 Evaluation Campaign: Overview of the Web People Search Clustering and Attribute Extraction Tasks. In *Third Web People Search Evaluation Forum (WePS-3), CLEF 2010*, Padua, Italy.

Bagga, A., Baldwin, B. 1998. Entity-based Cross-document Co-referencing Using the Vector Space Model. In *Proceedings of the 17th International Conference on Computational Linguistics, ACL*, pages 79-85, Montreal, Quebec, Canada.

Balog, K, He, J., Hofmann, K., et al. 2009. The University of Amsterdam at WePS2. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Banko, M., Cafarella, M. J., Soderland, S., Broadhead, M., Etzioni, O. 2007. Open Information Extraction from the Web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, pages 2670-2676, Hyderabad, India.

Bergsma, S., Lin, D., Goebel, R. 2009. Glen, Glenda or Glendale: Unsupervised and Semi-supervised Learning of English Noun Gender. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL)*, pages 120-128, Boulder, Colorado, USA.

Blei, D., Ng, A., Jordan, M. 2003. Latent Dirichlet Allocation. *Journal of Machine Learning Research*, 3, 993-1022.

Breiman, L. 1996. Bagging Predictors. *Machine Learning*, 24(2):123-140.

Brin, S. 1998. Extracting Patterns and Relations from the World Wide Web. In *WebDB Workshop at 6th International Conference on Extending Database Technology*, pages 172-183.

Bunescu, R., Mooney, R. J. 2005a. A Shortest Path Dependency Kernel for Relation Extraction. In *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pages 724-731, Vancouver, B.C., Canada.

Bunescu, R., Mooney, R. J. 2006. Subsequence Kernels for Relation Extraction. In Y. Weiss and B. Scholkopf and J. Platt (eds.), *Advances in Neural Information*

Processing Systems 18 (pp. 171-178). Cambridge, MA: MIT Press.

Bunescu, R.C., Pasca, M. 2006. Using Encyclopedic Knowledge for Named Entity Disambiguation. In *11th Conference of European Chapter of the Association for Computational Linguistics (EACL 2006)*, Trento, Italy.

Byrne, L., Dunnion, J. 2010. UCD IIRG at TAC 2010 KBP Slot Filling Task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Chen, Y., Lee, S. Y. M., Huang, C. 2009. PolyUHK: A Robust Information Extraction System for Web Personal Names. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Chen, Z., Tamang, S., Lee, A., Li, X., Lin, W., Snover, M., Artiles, J., et al. 2010. CUNY-BLENDER TAC-KBP2010 Entity Linking and Slot Filling System Description. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Collins, M. 2002. Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1-8, Philadelphia, USA.

Crammer, K., Dekel, O., Keshet, J., Shalev-Shwartz, S., Singer, Y. 2006. Online Passive-aggressive Algorithms. *Journal of Machine Learning Research*, 7, 551-585.

Cucerzan, S. 2007. Large-scale Named Entity Disambiguation Based on Wikipedia Data. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 708-716, Prague, Czech Republic.

Culotta, A. 2008. Learning and Inference in Weighted Logic with Application to Natural Language Processing. PhD thesis, University of Massachusetts.

Culotta, A., Kanani, P., Hall, R., Wick, M., McCallum, A. 2007. Author Disambiguation using Error-driven Machine Learning with a Ranking Loss Function. In *The Sixth International Workshop on Information Integration on the Web (IIWeb-07)*, Vancouver, Canada.

Culotta, A., McCallum, A. and Betz, J. 2006. Integrating Probabilistic Extraction Models and Data Mining to Discover Relations and Patterns in Text. In *Proceedings of the main Conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 296-303, New York, New York, USA.

Dang H. T., Owczarzak, K. 2009. Overview of the TAC 2009 Knowledge Base Population Track. In *Proceedings of TAC 2009 Workshop*, Gaithersburg, Maryland, USA.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R. 2004. The Automatic Content Extraction (ACE) Program Tasks, Data, and Evaluation. In *Proceedings of LREC*, pages 837-840, Lisbon, Portugal.

Dredze, M., McNamee, P., Rao, D., Gerber, A., Finin, T. 2010. Entity Disambiguation for Knowledge Base Population. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 277-285, Beijing, China.

Duchi, J. Hazan, E., Singer, Y. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121-2159.

Edmundson, H.P. 1969. New Methods in Automatic Extracting. *Journal of the ACM*, 16(2):264-285.

Ellis, J., Li, X., Griffit, K., Strassel, S. M., Wright, J. 2012. Linguistic Resources for 2012 Knowledge Base Population Evaluations. In *Proceedings TAC 2012 Workshop*, Gaithersburg, Maryland, USA.

Elmacioglu, E., Tan, Y. F., Yan, S., Kan, M., Lee, D. 2007. PSNUS: Web People Name Disambiguation by Simple Clustering with Rich Features. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 268-271, Prague, Czech Republic.

El-Beltagy, S.R. and Rafea, A. 2010. KP-Miner: Participation in SemEval-2. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL*, pages 190-193, Uppsala, Sweden.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A. M., Shaked, T., Soderland, S., Weld, D. S. and Yates, A. 2005. Unsupervised Named-entity Extraction from the Web: An Experimental Study. *Artificial Intelligence*, 165(1):91-134.

Farber, D. 2013. Microsoft's Bing Seeks Enlightenment with Satori. Cnet. CBS Interactive Inc. Retrieved 3 December 2013.

Fern, N., Fisteus, J. A., S, L., & Mart, E. 2010. WebTLab: A Co-occurrence-based Approach to KBP 2010 Entity-Linking Task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Frank, E., Paynter, G. W., Witten, I. H., Gutwin, C., Nevill-Manning, C. G. 1999. Domain-specific Keyphrase Extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence*, pages 668-673, Stockholm, Sweden.

Gabrilovich, E., Markovitch, S. 2007. Computing Semantic Relatedness using

Wikipedia-based Explicit Semantic Analysis. In *Proceedings of IJCAI 2007*, pages 6-12, Hyderabad, India.

Gong, J., Oard, D. 2009. Determine the Entity Number in Hierarchical Clustering for Web Personal Name Disambiguation. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Gooi, C. H., Allan, J. 2004. Cross-document Co-reference on a Large Scale Corpus. In *North American Chapter of the Association for Computational Linguistics (NAACL)*, Boston, Massachusetts.

Grishman, R., Sundheim, B. 1996. Message Understanding Conference-6: A Brief History. In *Proceedings of the 16th Conference on Computational Linguistics*, pages 466-471, Copenhagen, Denmark

Hammouda, K. M., Matute, D.N. and Kamel, M.S. 2005. CorePhrase: Keyphrase Extraction for Document Clustering. In *Proceedings of Machine Learning and Data Mining in Pattern Recognition, 4th International Conference, MLDM 2005*, pages 265-274, Leipzig, Germany.

Han, X., Zhao, J. 2009. NLPR_KBP in TAC 2009 KBP Track: A Two-Stage Method to Entity Linking. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA.

Han, X. and Zhao, J. 2009. CASIANED: Web Personal Name Disambiguation Based on Professional Categorization. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D. S. 2011. Knowledge-based Weak Supervision for Information Extraction of Overlapping Relations. In *Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 541–550, Portland, Oregon, USA.

Hulth, A., Megyesi, B. 2006. A Study on Automatically Extracted Keywords in Text Categorization. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the ACL*, pages 537-544, Sydney, NSW, Australia.

Hulth, A. 2003. Improved Automatic Keyword Extraction Given more Linguistic Knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, pages 216-223, Sapporo, Japan.

Ikeda, M., Ono, S., et al. 2009. Person Name Disambiguation on the Web by Two Stage Clustering. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Iria, J., Xia, L., Zhang, Z. 2007. WIT: Web People Search Disambiguation using Random Walks. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 480-483, Prague, Czech Republic.

Ji, H., Grishman, R., Dang, H. T., Griffitt, K., Ellis, J. 2010. Overview of the TAC 2010 Knowledge Base Population Track. In *Proceedings of TAC 2010 Workshop*, Gaithersburg, Maryland, USA.

Ji, H., Grishman, R., Dang, H. T. 2011. Overview of the TAC 2011 Knowledge Base Population Track. In *Proceedings of TAC 2011 Workshop*, Gaithersburg, Maryland, USA.

Ji, H., Lin, D. 2009. Gender and Animacy Knowledge Discovery from Web-scale N-grams for Unsupervised Person Mention Detection. In *Proceedings of the 23rd Pacific Asia Conference on Language, Information and Computation, PACLIC 23*, pages 220-229, Hong Kong, China.

Jordan, M.I., Ghahramani, Z., Jaakkola, T. S., Saul, L. K. 1999. An Introduction to Variational Methods for Graphical Models. *Machine Learning*, 37(2):183-233.

Kambhatla, N. 2004. Combining Lexical, Syntactic, and Semantic Features with Maximum Entropy Models for Extracting Relations. In *Proceedings of the ACL 2004*, Barcelona, Spain.

Kate, R. J., Mooney, R. J. 2010. Joint Entity and Relation Extraction using Card-pyramid Parsing. In *Proceedings of the Fourteenth Conference on Computational Natural Language Learning*, pages 203-212, Uppsala, Sweden.

Kim, S.N., Medelyan, O., Kan, M., Baldwin, T. 2010. SemEval-2010 Task 5: Automatic Keyphrase Extraction from Scientific Articles. In *Proceedings of SemEval-2: Evaluation Exercises on Semantic Evaluation*, Uppsala, Sweden.

Kozareva, Z., Vazquez, S., Montoyo, A. 2007. UA-ZSA: Web Page Clustering on the Basis of Name Disambiguation. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 338-341, Prague, Czech Republic.

Lafferty, J. D., McCallum, A. Pereira, F. C. N. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282-289, Williams College, Williamstown, MA, USA.

Lehmann, J., Monahan, S., Nezda, L., Jung, A., Shi, Y. 2010. LCC Approaches to Knowledge Base Population at TAC 2010. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Lefever, E., Fayruzov, T., Hoste, V., Cock, M. D. 2009. Fuzzy Ants Clustering for Web People Search. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Li, S., Gao, S., Zhang, Z., et al. 2009. PRIS at TAC 2009: Experiments in KBP Track. In *Proceedings of the Second Text Analysis Conference (TAC 2009)*, Gaithersburg, Maryland, USA.

Litvak, M., Last, M. 2008. Graph-based Keyword Extraction for Single-document Summarization. In *Proceedings of the Workshop on Multi-source Multilingual Information Extraction and Summarization*, pages 17–24, Manchester, UK.

Liu, Y., Shi, Z., Sarkar, A. 2007. Exploiting Rich Syntactic Information for Relation Extraction from Biomedical Articles. In *The Conference of the North American Chapter of the Association for Computational Linguistics*, pages 97-100, Rochester, New York, USA.

Long, C., Shi, L. 2010. Web Person Name Disambiguation by Relevance Weighting of Extended Feature Sets. In *Third Web People Search Evaluation Forum (WePS-3), CLEF 2010*, Padua, Italy.

Lopez, P., Romary, L. 2010. HUMB: Automatic Key Term Extraction from Scientific Articles in GROBID. In *Proceedings of the 5th International Workshop on Semantic Evaluation, ACL 2010*, pages 248-251, Uppsala, Sweden.

Matsuo, Y., Ishizuka, M. 2004. Keyword Extraction from a Single Document Using Word Co-occurrence Statistical Information. *International Journal on Artificial Intelligence Tools*, 13(1): 157-169.

McCallum, A., Li, W. 2003. Early Results for Named Entity Recognition with Conditional Random Fields, Feature Induction and Web Enhanced Lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 188-191, Edmonton, Canada.

McCallum, A., Schultz, K. and Singh, S. 2009. Factorie: Probabilistic Programming via Imperatively Defined Factor Graphs. In Y. Bengio, D. Schuurmans, J. Lafferty, C. K. I. Williams, and A. Culotta (eds.), *Advances in Neural Information Processing Systems 22*, 1249-1257.

McCallum, A., Wellner, B. 2004. Conditional Models of Identity Uncertainty with Application to Noun Co-reference. In *Neural Information Processing Systems (NIPS 2004)*, Vancouver, B. C., Canada.

McDonald, R., Pereira, F., Kulick, S., Winters, S., Jin, Y. and White, P. 2005. Simple Algorithms for Complex Relation Extraction with Applications to Biomedical IE. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 491-498, University of Michigan, USA.

Mcnamee, P. 2010. HLTCOE Efforts in Entity Linking at TAC KBP 2010. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Medelyan, O. 2009. Human-competitive Automatic Topic Indexing. PhD thesis. Department of Computer Science, University of Waikato, New Zealand.

Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing Order into Texts. In *Proceedings of EMNLP 2004*, pages 404-411, Barcelona, Spain.

Milgram, J., Cheriet, M., Sabourin, R. 2006. One Against One or One Against All: Which One is Better for Handwriting Recognition with SVMs? In *10th International Workshop on Frontiers in Handwriting Recognition (IWFHR 2006)*, La Baule, France.

Mintz, M., Bills, S., Snow, R., Jurafsky, D. 2009. Distant Supervision for Relation Extraction without Labeled Data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP*, pages 1003-1011, Suntec, Singapore.

Nguyen, T. D., Luong, M. 2010. WINGNUS: Keyphrase Extraction Utilizing Document Logical Structure. In *Proceedings of the 5th International Workshop* *on Semantic Evaluation*, pages 166-169, Uppsala, Sweden.

Nocedal, J. and Wright, S. J. 1999. Numerical Optimization. Springer.

Pantel, P., Fuxman, A. 2011. Jigs and Lures: Associating Web Queries with Structured Entities. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 83-92, Portland, Oregon, USA.

Peng, F., McCallum A. 2004. Accurate Information Extraction from Research Papers using Conditional Random Fields. In *Proceedings of Human Language Technology Conference and North American Chapter of the Association for Computational Linguistics (HLT-NAACL'04)*, pages 329-336, Boston, Massachusetts, USA.

Rabiner, L. R. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2):257-286.

Radford, W., Hachey, B., Nothman, J., Honnibal, M., Curran, J. R. 2010. Document-level Entity Linking: CMCRC at TAC 2010. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Rao, D., Garera, N., Yarowsky, D. 2007. JHU1: An Unsupervised Approach to Person Name Disambiguation using Web Snippets. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 199-202, Prague, Czech Republic.

Rao, D., McNamee, P., Dredze, M. 2011. Entity linking: Finding Extracted Entities in a Knowledge Base. In *Multi-source, Multi-lingual Information Extraction and Summarization*.

Ravichandran, D., Hovy, E. 2002. Learning Surface Text Patterns for a Question

Answering System. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 41-47, Philadelphia, PA, USA.

Riedel, S. Yao, L., McCallum, A. 2010. Modeling Relations and Their Mentions without Labeled Text. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases (ECML PKDD '10)*, pages 148-163, Barcelona, Spain.

Robert, C., Casella, G. 2004. Monte Carlo Statistical Methods. Springer.

Romano, L., Buza, K., Giuliano, C. 2009. XMedia: Web People Search by Clustering with Machinely Learned Similarity Measures. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Rosario, B., Hearst, M. A. 2004. Classifying Semantic Relations in Bioscience Text. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 430-437, Barcelona, Spain.

Roth, D., Yih, W. 2007. Global Inference for Entity and Relation Identification via a Linear Programming Formulation. In Lise Getoor and Ben Taskar (eds.), *Introduction to Statistical Relational Learning*. MIT Press.

Sha, F., Pereira, F. 2003. Shallow Parsing with Conditional Random Fields. In *Conference on Human Language Technology and North American for Computational Linguistics (HLT-NAACL)*, pages 213-220, Edmonton, Canada.

Sibson, R. 1973. SLINK: An Optimally Efficient Algorithm for the Single-Link Cluster Method. *The Computer Journal (British Computer Society)*, 16 (1): 30–34.

Singhal, A. 2012. Introducing the Knowledge Graph: Things, Not Strings. Official Blog (of Google). Retrieved May 18, 2012.

Smirnova, E., Avrachenkov, K., Trousse, B. 2010. Using Web Graph Structure for

Person Name Disambiguation. In *Third Web People Search Evaluation Forum (WePS-3), CLEF 2010*, Padua, Italy.

Steyvers, M., Griffiths, T. 2006. Probabilistic Topic Models. In *Latent Semantic Analysis: A Road to Meaning*, Laurence Erlbaum.

Sugiyama, K., Okumura, M. 2007. TITPI: Web People Search Task Using Semi-Supervised Clustering Approach. In *Proceedings of the 4th International Workshop on Semantic Evaluations (SemEval-2007)*, pages 318-321, Prague, Czech Republic.

Sun, A., Grishman, R., Xu, W., Min, B. 2011. New York University 2011 System for KBP Slot Filling. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA.

Surdeanu, M., McClosky, D., Tibshirani, J., Bauer, J., Chang, A. X., Spitkovsky, V. I., Manning, C. D. 2010. A Simple Distant Supervision Approach for the TAC-KBP Slot Filling Task. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland USA.

Surdeanu, M., Gupta, S., et al. 2011. Stanford's Distantly Supervised Slot-Filling System. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland USA.

Sutton, C., McCallum, A. 2006. An Introduction to Conditional Random Fields for Relational Learning. In Lise Getoor and Ben Taskar (eds.), *Introduction to Statistical Relational Learning*. MIT Press.

Teh, Y., Newman, D., Welling, M. 2006. A Collapsed Variational Bayesian Inference Algorithm for Latent Dirichlet Allocation. In *Neural Information Processing Systems*.

Tseng, H., Chang, P., Andrew, G., Jurafsky, D., Manning, C. 2005. A Conditional

Random Field Word Segmenter for Sighan Bakeoff 2005. In *Proceedings of the Fourth SIGHAN Workshop on Chinese Language Processing*, pages 168-171, Jeju Island, Korea.

Turney, P. 1999. Learning to Extract Keyphrases from Text. *Technical report*, National Research Council, Institute for Information Technology.

Varma, V., Bysani, P., Reddy, K., Reddy, V. B., Kovelamudi, S., Vaddepally, S. R., Nanduri, R., et al. 2010. IIIT Hyderabad in Guided Summarization and Knowledge Base Guided Summarization Track. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Venkateshan, P. 2009. Clustering Web People Search Results using Fuzzy Ant-based Clustering. In *2nd Web People Search Evaluation Workshop (WePS 2009), 18th WWW Conference*, Madrid, Spain.

Vishwanathan, S. V. N., Schraudolph, N. N., Schmidt, M. W., Murphy, K. P. 2006. Accelerated Training of Conditional Random Fields with Stochastic Gradient Methods. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 969-976, Pittsburgh, Pennsylvania, USA.

Wallach, H. 2002. Efficient Training of Conditional Random Fields. In *Proceedings of 6th Annual CLUK Research Colloquium*, University of Edinburgh.

Wan, X. and Xiao, J. 2008. CollabRank: Towards a Collaborative Approach to Single-Document Keyphrase Extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969-976, Manchester, UK.

Wick, M., Rohanimanesh, K., Culotta, A., McCallum, A. 2009. Samplerank: Learning Preferences from Atomic Gradients. In *Neural Information Processing Systems (NIPS), Workshop on Advances in Ranking*.

Wick, M., Singh, S., McCallum, A. 2012. A Discriminative Hierarchical Model for Fast Coreference at Large Scale. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, pages 379-388, Jeju Island, Korea.

Wick, M., Kobren, A., McCallum, A. 2013. Large-scale Author Co-reference via Hierarchical Entity Representations. In *Proceedings of the 30th International Conference on Machine Learning*, Atlanta, Georgia, USA.

Witten, I.H., Paynter, G.W., Frank, E., Gutwin, C., Nevill-Manning, C.G. 2000. KEA: Practical Automatic Keyphrase Extraction. In *Working Paper 00/5*, Department of Computer Science, University of Waikato, Hamilton, New Zealand.

Wu, Z., Giles, C. L. 2013. Measuring Term Informativeness in Context. In *Proceedings of NAACL-HLT 2013*, pages 259-269, Atlanta, Georgia, USA.

Xu, J. Lu, Q. and Liu Z. 2012. Combining Classification with Clustering for Web Person Disambiguation. In *Proceedings of the 21st International Conference Companion on World Wide Web*, pages 637-638, Lyon, France.

Xu, J., Liu, Z., Lu, Q., Liu, P., Wang, C. 2011. Polyucomp in TAC 2011 Entity Linking and Slot-Filling. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA.

Xu, J., Lu, Q., Liu, Z. 2012. Aggregating Skip Bigrams into Key Phrase-based Vector Space Model for Web Person Disambiguation. In *Proceedings of KONVENS 2012*, pages 108-117, Vienna, Austria.

Yedidia, S., Freeman, W.T., Weiss, Y. 2004. Constructing Free Energy Approximations and Generalized Belief Propagation Algorithms. *Technical Report TR2004-040*, Mitsubishi Electric Research Laboratories.

Zelenko, D., Aone, C., Richardella, A. 2003. Kernel Methods for Relation Extraction.

Journal of Machine Learning Research, 3:1083-1106.

Zha, Hongyuan. 2002. Generic Summarization and Key Phrase Extraction Using Mutual Reinforcement Principle and Sentence Clustering. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval Tampere*, pages 113-120, Tampere, Finland.

Zhang C, Wang H., Liu, Y. Wu Dan, et.al. 2008. Automatic Keyword Extraction from Documents Using Conditional Random Fields. *Journal of Computational Information Systems*, 4(3):1169-1180.

Zhang, W., Tan, C. L., Su, J., Chen, B., et al. 2011. I2R-NUS-MSRA at TAC 2011: Entity Linking. In *Proceedings of the Fourth Text Analysis Conference (TAC 2011)*, Gaithersburg, Maryland, USA.

Zhang, W., Tan, C. L., Sim, Y.C., J. Su. 2010. NUS-I2R: Learning a Combined System for Entity Linking. In *Proceedings of the Third Text Analysis Conference (TAC 2010)*, Gaithersburg, Maryland, USA.

Zhao, H., Kit, C. 2011. Integrating Unsupervised and Supervised Word Segmentation: The Role of Goodness Measures. *Information Sciences*, 181(1):163–183.

Zhu, X., Ghahramani, Z. 2002. Learning from Labeled and Unlabeled Data with Label Propagation. *Technical Report CMU-CALD-02-107*, Carnegie Mellon University.

Appendix 1 List of Verified Entities in WePS2

This appendix gives the number of entities in Freebase and WePS2 dataset, the entity matches between the two of them. The entity links to Wikipedia articles are listed as well. "*id*" refers to the entity identifier in the WePS2 dataset.

	#Freebase Entities	# Answer Entities	Match between Freebase & Answer	Links to Wikipedia Entries
Amanda Lentz	0	20	0	N/A
Benjamin Snyder	1	28	0	N/A
Bertram Brooker	1	1	1	1. id=1: http://en.wikipedia.org/wiki/Bertram_Brooker
Cheng Niu	0	7	0	N/A
David Tua	5	1	1	1. id=1: http://en.wikipedia.org/wiki/David_Tua
David Weir	7	26	4	1. id=19: http://en.wikipedia.org/wiki/David_Weir_(athlete) 2. id=16: http://en.wikipedia.org/wiki/David_Weir_(academic) 3. id=3: http://en.wikipedia.org/wiki/David_Weir_(journalist) 4. id=1: http://en.wikipedia.org/wiki/David_Weir_(Scottish_footballer)
Emily Bender	0	19	0	N/A
Franz Masereel	0	3	0	N/A
Gideon Mann	0	2	0	N/A
Hao Zhang	2	24	1	1. id=1: http://en.wikipedia.org/wiki/Hao_Zhang
Helen Thomas	11	3	1	1. id=1: http://en.wikipedia.org/wiki/Helen_Thomas
Herb Ritts	8	2	2	1. id=2: N/A 2. id=1: http://en.wikipedia.org/wiki/Herb_Ritts
Hui Fang	3	21	1	1. id=7: http://en.wikipedia.org/wiki/Hong_Huifang
Ivan Titov	1	5	1	1. id=1: N/A
James Patterson	26	4	1	1. id=1: http://en.wikipedia.org/wiki/James_Patterson
Janelle Lee	0	34	0	N/A
Jason Hart	9	22	4	1. id=1: http://en.wikipedia.org/wiki/Jason_Hart_(basketball) 2. id=7: http://en.wikipedia.org/wiki/Jason_Hart_(baseball) 3. id=4: http://en.wikipedia.org/wiki/Jason_Hart_(musician) 4. id=8: N/A

Jonathan Shaw	7	26	3	1. id=3: http://en.wikipedia.org/wiki/Jonathan_Shaw_(tattooist) 2. id=2: http://en.wikipedia.org/wiki/Jonathan_Shaw_(politician) 3. id=5: http://en.wikipedia.org/wiki/Jonathan_Shaw_(photographer
Judith Schwartz	0	30	0	N/A
Louis Lowe	1	24	1	1. id=1: http://en.wikipedia.org/wiki/Enoch_Louis_Lowe
Mike Robertson	7	39	2	1. id=3: http://en.wikipedia.org/wiki/Mike_Robertson_(baseball) 2. id=8: http://en.wikipedia.org/wiki/Mike_Robertson_(snowboarder
Mirella Lapata	0	2	0	N/A
Nicholas Maw	2	1	1	1. id=1: http://en.wikipedia.org/wiki/Nicholas_Maw
Otis Lee	2	26	2	1. id=1: http://en.wikipedia.org/wiki/Otis_Lee_Crenshaw#Otis_Lee_Crenshaw 2. id=6: http://en.wikipedia.org/wiki/Otis_Lee_Birdsong
Rita Fisher	1	24	1	1. id=3: N/A
Sharon Cummings	0	30	0	N/A
Susan Jones	13	56	3	1. id=31: N/A 2. id=15: N/A 3. id=10: N/A
Tamer Elsayed	0	8	0	N/A
Theodore Smith	1	54	0	N/A
Tom Linton	3	10	2	1. id=1: http://en.wikipedia.org/wiki/Jimmy_Eat_World 2. id=12: N/A
Total	111	552	32	

Appendix 2 Trigger Words List

This appendix gives the manually compiled trigger words for Wikipedia relation types in Chapter 6 Person Profile Extraction.

Relation Types	Trigger Words
person.birthplace	birth, birthplace, born
person.deathplace	committed suicide, dead, death, deathplace, died, executed, fatally wounded, guillotined, killed, mortally wounded, murdered, shot dead, strangled

person.sibling	brother, brother-in-law, brother-sister, brothers, brothers-in-law, cousin, cousins, half-brother, half-brothers, half-cousin, half-cousins, half-siblings, half-sister, half-sisters, half-twin, sibling, siblings, sister, sister-in-law, sisters, sisters-in-law, stepbrother, stepbrothers, stepsister, stepsisters, twin, twins
person.place_lived	grew, grow, grows, growing, grew up, grow up, growing up, grown up, grows up, live, lived, lives, living, move, moved, moves, moving, reside, resided, resides, residing
person.nationality	nationality, ethnicity, citezein, citizen, citizenry, citizens, citizenship, descendent, ethnic, ethnic background, ethnically, ethnicities, ancestry, bloodline, civic duty, ethnic classification, ethnic group, ethnic groups, native, ethnic identity, ethnic origin, ethnick, ethnos, honorary citizen, honorary citizenship, national origin, nation, nationalities, origin, origination, pedigree, people group, people groups, rootage, homeland, motherland
person.marriage	married, marriage, wife, husband
person.parents_children	son, sons, daughter, daughters, child, children, boy, boys, girl, girls, mother, father, parent, parents, stepson, stepdaughter, stepmother, stepfather
person.education	graduate, graduated, graduation, graduating, postgraduate, undergraduate, certificate, diploma, educated, educate, education, educating, PhD, BS, MS, bachelor, doctor, master, student, students, bachelors, masters, pupil
organization.founders	established, founded, launched, founder, founders, co-founder, co-founders
organization.parent_child	subsidiary, acquired, acquisition, owned, takeover
organization.headquarters	based in, head office, headquarter, headquartered in, headquarters, headquartered
location.capital_of	Capital
country.administrative_divisions	bailiwick, bailiwicks, borough, boroughs, canton, cantons, city, cities, commune, communes, county, counties, district, districts, duchy, duchies, emirate, emirates, federal state, federal states, municipality, municipalities, parish, parishes, prefecture, prefectures, province, provinces, region, regions, rural district, rural districts, shire, shires, subdistrict, subdistricts, town, towns, township, townships, village, villages, viceroyalty, viceroyalties, voivodeship, voivodeships, palatinatum, palatinatums, division, divisions, subdivision, subdivisions, states, village, villages

business.employment_tenure	CEO, CFO, actor, adjunct professor, admiral, adviser, advisor, anchor, animator, anthropologist, archaelogist, assistant, assitant professor, associate director, associate professor, astrophysicist, balletmaster, balletmaster-in-chief, biologist, captain, chair, chaired, chairman, chairperson, chancellor, chief content officer, chief executive officer, chief financial officer, chief operating officer, choreographer, co-director, co-editor, coach, collaborator, programmer, constable, coordinator, correspondent, correspondents, criminologist, cryptographer, curator, dean, designer, designers, diplomat, diplomats, directed, director, director-general, directorship, economist, editor, editor in chief, editor-in-chief, educator, employee, employees, engineering manager, essayist, executive, executive director, executives, expert, faculty, field coach, film producer, general manager, governance fellow, governor, head, headed, headmaster, historian, hosted, host, hosts, hosting, industrial designer, inspector, inspector general, instuctor, investment manager, investor, journalist, lecturer, lieutenant general, lieutenant-general, manager, managing director, managing partner, mathematician, minister, musician, musicologist, neuroscientist, news anchor, newscaster, novelist, officer, painter, paleontologist, philosopher, physicist, police chief, president, principal, professor, reporter, research scholar, scientist, sculptor, secretary general, secretary-general, senior engineer, senior lecturer, senior vice-president, sergeant, serving as, shooting guard, sinologist, sociologist, studied under, supervised by, taught, teachers, teaches, teaching, television anchor, then-secretary-general, theorist, under supervision of, under the direction of, under the supervision of, under the tutelage of, venture capitalist, vice chancellor, vice dean, vice president, vice-president, worked, working, works