Named Entity Translation Matching and Learning with Mining from Multilingual News

CHEUNG Pik Shan
張碧珊

香港中文大學
THE CHINESE UNIVERSITY OF HONG KONG

A Thesis
Submitted in Partial Fulfilment of the Requirements for the Degree of
Master of Philosophy
in
Systems Engineering and Engineering Management

June 2004

Abstract

We propose a novel named entity matching model which considers both semantic and phonetic clues. The matching is formulated as an optimization problem. One major component is a phonetic matching model which exploits similarity at the phoneme level. It can handle new or unseen names by considering the information at the phoneme level. We investigate three learning algorithms for obtaining the similarity information of basic phoneme units based on training examples. By applying this proposed named entity matching model, we also develop a mining framework for discovering new, unseen named entity translations from online daily Web news. This framework harvests comparable news in different languages using an existing bilingual dictionary. It is able to discover new name translations not found in the dictionary. Experiments show that our proposed matching model can handle named entity matching in a more flexible way. Name translations not found in the dictionary can be effectively discovered from daily news.

摘要

我們提出了一個嶄新的專有名詞配對模型，該模型同時利用語義及語音線索，此配對模型被制定成一個最優化課題。其中一個主要的元件是利用了音位層次相似度的語音配對模型，它可以透過利用音位層次的資訊來處理全新或未知的名詞。我們探究了三個訓練演算法，本著訓練樣本而得出基本音位單位的相似度資訊。我們亦應用此專有名詞配對模型而發展出一個採擷架構，在每日網上新聞中找出新的或是未知的專有名詞翻譯。這個架構利用已有的雙語辭典去採集不同語言的相似新聞，它能夠找出一些在辭典中找不到的新名詞翻譯。實驗証明我們提出的配對模型可以利用一個較有彈性的方式去處理專有名詞配對問題，在辭典中找不到的名詞翻譯亦可以有效地在每日的新聞中被找出來。

Acknowledgments

This thesis is the result of my two years of work whereby I have been accompanied and supported by many people. It is a pleasant aspect that I have the opportunity to express my gratitude for all of them.

The first person I would like to thank is my supervisor Dr. Wai Lam. He has been my supervisor since I was a final year student during my undergraduate study. He has taught me a lot on research mind and problem solving skills. Thanks for his patience and time spent on me. Without his guidance, I would be like a sheep without a rudder.

Besides, I would like to thank my pretty partner, Kaka, who is a nice girl to get along with. We helped each other in many aspects, including tutorial and research work. I wish her every success in her Ph.D. study.

I wish to thank my friends, who have offered me full support and help when I was in trouble. Jam is the smartest guy I have ever met. I am glad to have him by my side during my M.Phil. life. Stanley, who is a real genius and

my greatest mentor. Thank you for his continuous feeding of mental food. The food has been keeping me work lively. Polly, Cecilia, Gabriel, Gatien, and Ah Ki are always willing to offer help on my research issues without hesitation. In my hardest period of the Master study, they gave me energy to continue my work. They shared my joy as well as sadness.

My family has given me endless support since I was born. No matter what decision I make, they give full trust on me. I always try my best to be a good girl because of my family. I promise that when I meet my father again, he will be proud of having such a daughter.

These two years is a turning point in my life. I have learnt to be a confident person and I have explored my potential and my future. I cherish the time I have spent in CUHK and will never forget my Master life.

Lastly, I would like to say Thank you to my dearest God, who offered me this opportunity to study the Master degree. He is the one who makes things HAPPEN. He has given me more than I have ever expected and I aspire to dedicate whatever I have to Him. Although my future is not known, with the Savior stands beside me, I have nothing to fear.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Many existing systems dealing with cross-language documents make use of bilingual dictionaries. In all these systems, a fixed dictionary is used throughout the process implying that only those terms exist in the dictionary can be handled. Obviously, these systems encounter difficulties when they process new or unseen terms which are common especially for named entities. A study has shown that many terms in the submitted queries for news search consist of named entities or proper nouns [22]. One promising approach is to discover new, unseen bilingual name translations automatically. In this thesis, we propose a novel named entity matching model to match bilingual named entities automatically. By applying this proposed named entity matching model, we also develop a mining framework for discovering new, unseen named entity translations from online daily Web news.

In the following sections, we will briefly introduce the two main areas of our work, namely, named entity translation matching and mining new named entity translations from news. Then we will discuss the organization of this thesis.

1.1 Named Entity Translation Matching

Translations of many named entities involve both semantic meaning and pronunciation. We investigate a model which considers both semantic and phonetic clues given two bilingual entities. The matching is formulated as an optimization problem. One major component in our named entity matching model is a phonetic matching model which exploits phonetic information. In particular, it considers similarity at the phoneme level. The similarity information of basic phoneme units between English and Chinese phonemes is captured in a phoneme pronunciation similarity (PPS) table. We investigate three learning algorithms for obtaining the similarity values of the PPS tables based on a set of training data. The three learning algorithms include Widrow-Hoff (WH) algorithm, Exponentiated-Gradient (EG) algorithm , and the Genetic algorithm. The experiment results show that the learned similarity values are better than the manaually assigned ones. Our new named entity translation matching model and the learning algorithms

have been published in ACM SIGIR Conference 2004 [17].

1.2 Mining New Translations from News

We apply the named entity matching model in our multilingual news mining framework. Online daily Web news are widely available from different agencies or subscribed delivery services. Many events arise everyday and they are typically reported in different news sources. It is very common that a particular event is covered by multiple stories from different sources. An event is defined as a specific piece of incident or activity which usually occurs in a short period of time. For example, "A worm, reportedly called Slammer, halted Internet traffic over the weekend of January 26 2003 in some parts of the world" is a sample event. Typically, when a new event arises, news from different sources in different languages may report the same event. The progress of the event will also trigger news in subsequent days. If different sources offer news in different languages, a cross-language comparable news cluster reporting the same event can be collected. The usage of names in a comparable news cluster provides good clues on the discovery of name translations. Often, some names are new or unseen; thus they cannot be found in an existing bilingual lexicon.

We have developed a novel online daily Web news mining system which

possesses the above characteristics. News documents are automatically fetched from different Web sources. Some sources offer English news while some provide Chinese news. Our mining system is able to discover unseen named entity tranlsations on a daily basis. Our mining framework has been published in the International Conference on Information Technology 2004 [6] and the Joint Conference on Digital Libraries 2004 [16].

1.3 Thesis Organization

The remaining chapters of the thesis are organized as follows. Chapter 2 presents related work on term translation mining approach. Chapter 3 discusses our approach for named entity matching. Chapter 4 describes the phonetic matching model. In Chapter 5, we focus on the learning of phonetic similarity information. Experiments on named entity matching model are presented in Chapter 6. In Chapter 7, our approach for mining new named entity translations from news is presented. Chapters 8 and 9 discuss the experiments on mining new translations and context-based gloss translation respectively. In the last chapter, we draw the conclusions and mention some future work.

Chapter 2

Related Work

There were some related work on extracting term translations. One early work on automatic identification of word translations from nonparallel corpus was presented in [21]. It mainly makes use of co-occurrence statistics. An algorithm using context seed word and term statistics was designed to extract bilingual lexicon from nonparallel, comparable corpus [8]. A method called Convec was developed to generate bilingual lexicon from comparable corpus [7]. This model employs information retrieval techniques and makes use of the context of unknown words in the source and target languages.

An attempt for mining term translations from Web anchor texts was investigated in [19]. They discovered that anchor texts linking to the same page may contain terms with similar semantics, and that some of them may be written in different languages. Therefore, a candidate term has a higher

chance of being effectively translated if it is written in the target language and frequently co-occurs with the source query term. However, this approach was restricted to discover those terms appeared in the anchor texts.

Nie et al. proposed a technique for mining parallel documents from parallel Web sites [20]. An alignment model was proposed to generate parallel sentences for cross-lingual information retrieval. Mining parallel sentences from a bilingual comparable news collection was proposed in [26]. Sentences from a bilingual news collection were aligned based on an alignment model. Their model made use of iterative EM algorithm to calculate the translation probabilities of sentences. The aligned parallel sentence pairs were then used to train a word alignment model.

Huang et al. presented an integrated approach to extract a named entity translation dictionary from a bilingual corpus [10]. A statistical alignment model was used to align the named entities. An iterative process was applied during the entity extraction. In their other related work [12], an automatic extraction of Hindi-English (H-E) named entities from bilingual parallel corpora was developed. This model is first trained by the extracted C-E named entity pairs. Then the H-E named entity pairs are iteratively updated based on newly extracted ones. A list of named entity pairs with minimum transliteration cost are chosen.

One major drawback of the above approaches is that they do not consider

phonetic information. A similarity-based backward transliteration approach was proposed in [18] to automatically acquire phonetic similarities. A shortcoming of this approach is that it does not take into account semantic information. However, many named entity translations involve both semantic and phonetic information at the same time. In the recent work of Huang et al. [11], named entity tagging cost, transliteration cost and word-based translation cost were all considered to extract the best named entity pairs. Both phonetic and semantic features are considered in this model. Dynamic programming is applied in searching for the optimal alignment between English letters and pinyin syllables. However, a parallel corpus is still required for this approach.

A major advantage of our proposed approach over the above existing methods is that our approach analyzes both semantic and phonetic information and formulates the problem as a number of optimization models. Our approach does not need a parallel corpus. The basic phonetic similarity information can be obtained via a learning process. For our framework for mining new named entity translations from news, one feature is that comparable news are automatically detected to facilitate new translation discovery.

For mining new named entity translations from news, one major component of our framework is on term disambiguation of the translation component. In dealing with the selection of good translations from a set of terms,

several studies has been done on utilizing mutual information between terms. Gao et al. [9] applied the co-occurrence model on query translations. Our gloss translation model shares some resemblances with their model with appropriate modifications. As they just need to translate a query (a single sentence), our challenge is bigger. We have to deal with the whole content of the news article implying a much higher computational cost when considering the co-occurrence model. Chen and Ku [5] applied the mutual information model on term disambiguation. However, they only selected one translation among all the English terms retrieved while we preserve most of the translations by a score-based weighting scheme. Those translations with score lower than a certain threshold are filtered. The reason for retaining translations with high weights is that preserving more reasonable translations will maximize the performance of cross-language comparable news detection. Moreover, our model considers more sophisticated distance factor and cohesion with neighboring terms in the formulation.

Chapter 3

Named Entity Matching Model

3.1 Problem Nature

The objective of our named entity matching model is to compute the similarity between two given named entities written in two languages. Note that this is a different problem from cross-language transliteration. Cross-language transliteration attempts to generate the translation of a term in one language given a term in another language. However, in this named entity matching problem, we attempt to compute a kind of similarity between two given entities in two languages.

Given a pair of named entities which are translation of each other, it is common to find part of the entity is matched based on semantic and the remaining part is based on phonetic clues. For example, consider the English

entity "University of Akron" and its corresponding translated Chinese entity "阿克倫大學". If we just adopt the semantic clue, we can match the term "University" with "大學" based on a bilingual dictionary. However, "Akron" vs "阿克倫" will be missed as they may not be found in a typical dictionary. On the contrary, if we just consider the phonetic clues, "Akron" can match with "阿克倫" due to the similarity of their pronunciations but "University" and "大學" do not match.

In general, there are five issues that need to be addressed in named entity matching. The first issue is that we need to consider both semantic and phonetic clues when dealing with the matching of two named entities in different languages. The example stated above is a typical one with both semantic and phonetic clues mixed in a single named entity.

The second issue that needs to be addressed is the capability of handling unseen names. For example, the term "Kadyrov" was the name of the former president of Chechen who was killed in a bomb in the middle of May 2004. This name cannot be found in a typical bilingual dictionary. For phonetic matching, out-of-vocabulary terms may exist in either language. We investigate the generation of the phonetic representation by a machine learning technique.

The third issue is related to the tokenization of the entities. For a given entity in Chinese, we need to break it into appropriate tokens to facilitate the

matching. For a given entity in English, it is usually composed of separate terms. However, some terms may need to be grouped in order to be effectively mapped to Chinese terms phonetically.

The fourth issue is related to the matching of tokens not necessarily in sequence. In the example of "University of Akron" and its corresponding translated Chinese entity "阿克倫大學", the first token "University" should match with the second token of Chinese entity "大學" while the third English token "Akron" should match with the first Chinese token "阿克倫". Therefore, we need to consider any possible sequences of matching when dealing with the named entity matching procedure.

The fifth issue is to consider partial matching. For example, consider the entity "Palo Alto Chamber of Commerce" and its corresponding translated Chinese entity "帕洛阿爾托商會". After looking up the bilingual dictionary, translations for the English terms "Chamber" and "Commerce" can be found. In particular, "Chamber" can match with the Chinese word segment "會" in the Chinese entity. One of the translations found for the term "Commerce" is "商業". Although this translation cannot be fully matched in the Chinese entity, it can be partially mapped to the Chinese entity via the term "商". Other translations of the term "Commerce" cannot be mapped to any word segments in the Chinese entity at all.

3.2 Matching Model Investigation

To tackle the above issues, we investigate a named entity matching model which analyzes both semantic and phonetic similarities between different tokens of the named entity pair. The semantic mapping is mainly determined based on the bilingual dictionary. The phonetic similarity is handled by the phonetic matching model described in Chapter 4.

Consider an English entity E represented by terms $<t_1, \ldots, t_{m_0}>$ and a Chinese entity C represented by Chinese characters $<s_1, \ldots, s_{n_0}>$. For each English term t_i, the bilingual dictionary is looked up. The current bilingual dictionary we use is derived from the one provided by Linguistic Data Consortium (LDC) with additional entries inserted manually. Typically, a set of Chinese translations are found in the dictionary for t_i. The Chinese entity C is scanned to get those word segments which can fully or partially match with any of the Chinese translations. Therefore, the term t_i may map to some Chinese word segments. Each word segment is composed of consecutive Chinese characters. Let the matched word segments be represented as $((d^{t_i}_{1,F}, d^{t_i}_{1,L}), ((d^{t_i}_{2,F}, d^{t_i}_{2,L}), \ldots)$ where $d^{t_i}_{j,F}$ and $d^{t_i}_{j,L}$ denote the starting and ending position of the Chinese character respectively in C for the j-th matched word segment. In other words, $(d^{t_i}_{j,F}, d^{t_i}_{j,L})$ corresponds to certain consecutive Chinese characters $(s_{k_1}, s_{k_1+k_2})$. There is also a weight associated with each

word segment reflecting the degree of matching. Similarly, an English term may be able to match with certain word segments in the Chinese entity C phonetically. For example, the term "Alto" can map to the word segment "阿爾托" based on phonetic evidence. In general, for an English term t_i, it can match with some word segments in C phonetically. Let the phonetically matched word segments be represented as $((p^{t_i}_{1,F}, p^{t_i}_{1,L}), (p^{t_i}_{2,F}, p^{t_i}_{2,L}), \ldots)$ where $p^{t_i}_{j,F}$ and $p^{t_i}_{j,L}$ denote the starting and ending position of the Chinese character respectively in C for the j-th matched word segment. There is also a weight associated with each word segment reflecting the degree of phonetic matching. Figure 3.1 shows a diagram illustrating the modeling.

	s_1 s_2	
t_1	$(d^{t_1}_{1,F} \cdots d^{t_1}_{1,L})$	$(d^{t_1}_{2,F} \cdots d^{t_1}_{2,L})$
	$(p^{t_1}_{1,F} \cdots p^{t_1}_{1,L})$	
t_2	$(d^{t_2}_{1,F} \cdots d^{t_2}_{1,L})$	$(d^{t_2}_{2,F} \cdots d^{t_2}_{2,L})$
	$(p^{t_2}_{1,F} \cdots p^{t_2}_{1,L})$	$(p^{t_2}_{2,F} \cdots p^{t_2}_{2,L})$
t_3	:	
	:	

Figure 3.1: The modeling of the named entity matching

Armed with the above modeling, the objective is to find a set of mapping between English terms and Chinese word segments such that the total

weight is maximized subject to constraints that each English term can only be mapped at most once and the mapped Chinese word segments cannot be overlapping. Let $((q_F^{t_{i_1}}, q_L^{t_{i_1}}), (q_F^{t_{i_2}}, q_L^{t_{i_2}}) \ldots)$ denote a particular solution. Each $(q_F^{t_j}, q_L^{t_j})$ can be either $(d_{k,F}^{t_j}, d_{k,L}^{t_j})$ or $(p_{k,F}^{t_j}, p_{k,L}^{t_j})$. Each t_j can only appear one time and each word segment represented by q cannot be overlapping.

In principle, the solution for the maximization problem can be found using an exhaustive search. However, the complexity is very high in practice especially when a large number of English and Chinese entity pairs need to be evaluated in a timely fashion as exemplified by the mining framework for discovering new, unseen translations from daily news described in the later part of this paper. After taking all the factors into consideration, we develop a named entity matching model which can address all the issues mentioned above and possess reasonable computational complexity. This named entity matching model is composed of two tasks. The first task is to conduct tokenization on both English and Chinese named entities. The second task is to make use of a hybrid semantic and phonetic matching algorithm which formulates the problem as a bipartite weighted graph problem. These two tasks of the named entity matching model are presented in detail below.

3.3 Tokenization

The terms appeared in the named entities need to be organized into appropriate tokens to facilitate the matching. The tokenization process of both English and Chinese entities is performed based on the bilingual dictionary. Consider a pair of English and Chinese named entities. Each term in the English named entity is looked up in the bilingual dictionary. Typically, a set of Chinese translations are found the dictionary for a particular English term. The Chinese entity is scanned to get those word segments which can maximally match with one of the Chinese translations. If the degree of this maximal matching exceeds or reaches a certain threshold θ_t, the English term as well as those Chinese word segments are treated as separate tokens. The degree of matching ρ is defined as the number of matched characters divided by the total number of characters in the corresponding term of the Chinese translation. Returning to the above example, the term "Commerce" matches with the term "商" with $\rho = 0.5$. The English terms "Chamber" and "Commerce" as well as the Chinese segments "會" and "商" are treated as separate tokens.

The next step of the tokenization process is to group adjacent terms which do not involve in the dictionary mapping. These adjacent terms are concatenated to form a single token to facilitate possible mapping due to phonetic

similarity. Returning to the above example, the English terms "Palo" and "Alto" will be concatenated to form a token "Palo Alto". The Chinese segment "帕洛阿爾托" will be treated as a single token. As a result, the English entity will be broken into four tokens, namely, "Palo Alto", "Chamber", "of", and "Commerce" whereas the Chinese entity will be broken into three tokens, namely, "帕洛阿爾托", "商", and "會".

3.4 Hybrid Semantic and Phonetic Matching Algorithm

After tokenization, the entities are represented by a sequence of tokens. Let the English entity, E, be represented as tokens $<e_1, \ldots, e_m>$ and the Chinese entity, C, be represented as tokens $<c_1, \ldots, c_n>$. Since our objective is to conduct a mapping of tokens between English and Chinese, we can formulate the matching problem via an undirected bipartite weighted graph. Each token is associated with a graph vertex. Let V be the vertex set and L be the edge set. The vertex set V is set to $\{V_E \cup V_C\}$ where $V_E = \{e_1, \ldots, e_m\}$ and $V_C = \{c_1, \ldots, c_n\}$. If there is a mapping found semantically or phonetically between an English token e_i and a Chinese token c_j, there will be an edge $(e_i, c_j) \in L$.

The weight $\mu(e_i, c_j)$ of the edge is determined by the degree of mapping of the associated tokens. Generally, semantic mapping is relatively more reliable than phonetic mapping. The edge construction process starts with considering the semantic mapping using the bilingual dictionary as described in the tokenization process above. If there is a mapping found with sufficient degree, an edge (e_i, c_j) is formed with the weight $\mu(e_i, c_j)$ set to ρ as defined in the tokenization process. It can be easily shown that $\theta_t \leq \rho \leq 1$. Next, we consider phonetic mapping between tokens. For each English token e_i, which does not have semantic mapping with Chinese tokens, we compute the phonetic similarity, $\mu(e_i, c_j)$ between e_i and each Chinese token, c_j. The phonetic similarity is calculated using our phonetic matching model described below. The range of this phonetic similarity value is $(0, 1]$. If the phonetic similarity is larger than zero, an edge is constructed with the phonetic similarity assigned to the weight of the edge.

After the edges and weights of the graph have been constructed, it is obviously a bipartite weighted graph. The matching problem is reduced to finding a set of edges such that the total weight Ω is maximized and each token can only be mapped to a single token on the other side. Therefore, this requirement can be formulated as a bipartite weighted graph matching

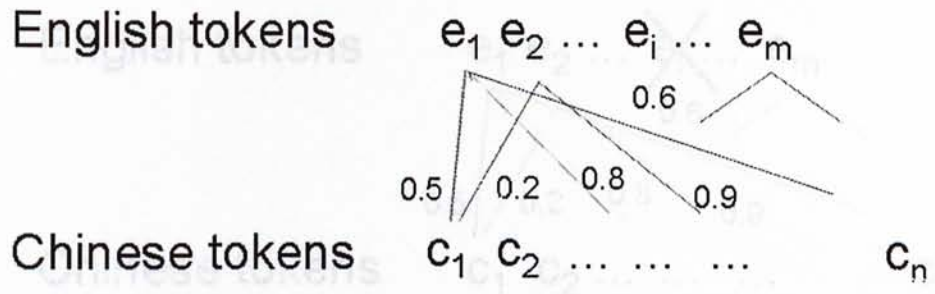

Figure 3.2: Sample named entity matching modeled by a bipartite weighted graph matching

problem [1]. Formal description of the problem is given as follows:

$$\text{Maximize } \Omega = \sum_{(e_i,c_j)\in L} \mu(e_i, c_j)x(e_i, c_j) \tag{3.1}$$

subject to

$$\sum_{c_j:(e_i,c_j)\in L} x(e_i, c_j) = 1 \quad \forall e_i \in V_E$$

$$\sum_{e_j:(e_j,c_i)\in L} x(e_j, c_i) = 1 \quad \forall c_i \in V_C$$

$$x(e_i, c_j) \geq 0 \quad \forall (e_i, c_j) \in L$$

where $x(e_i, c_j)$ is a binary variable representing whether the mapping between e_i and c_j is chosen in the solution. Figure 3.2 shows a sample named entity matching between English and Chinese tokens modeled by a bipartite weighted graph matching. To solve the maximization problem, one can formulate it as a minimum cost assignment problem [1]. The minimum cost assignment problem attempts to find a mapping for every token.

We conduct a series of transformation on the bipartite graph in order to

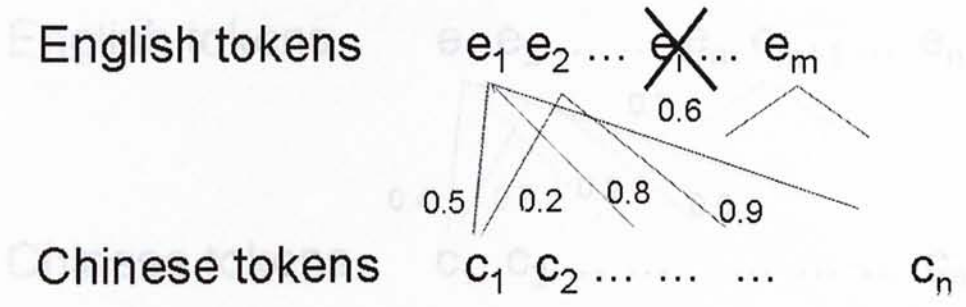

Figure 3.3: Removal of tokens not associated with any edges in the original graph

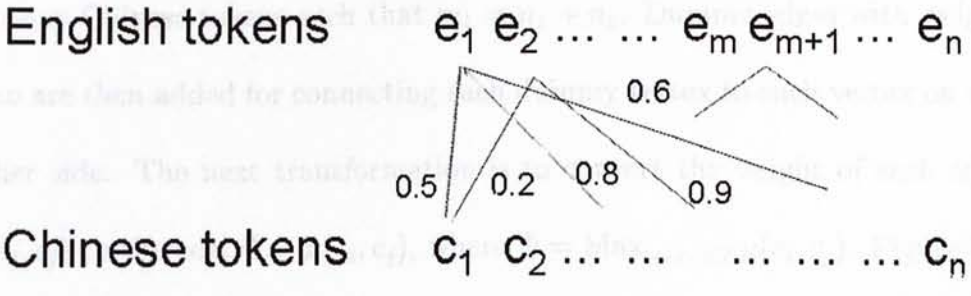

Figure 3.4: Construction of dummy vertices

fulfill the requirement of the cost assignment problem. The first transformation is to remove those tokens which do not associate with any edges in the original bipartite graph as shown in Figure 3.3. Suppose m_1 and n_1 is the number of vertices for English and Chinese tokens respectively after this vertex removal step. Then, we construct some dummy vertices to balance the number of vertices between the English and Chinese tokens as shown in Figure 3.4. For example, if $m_1 > n_1$, we add n_k vertices representing

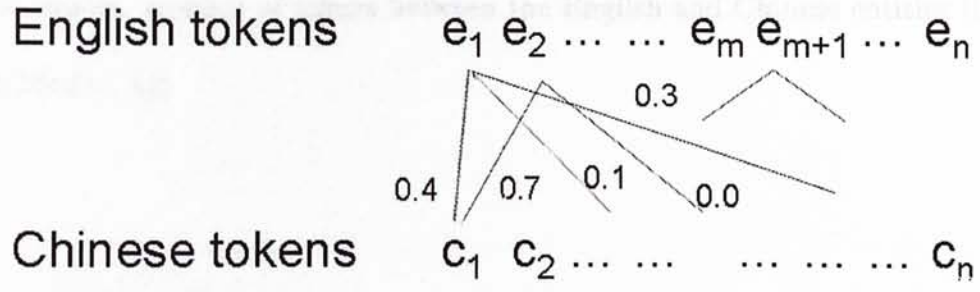

Figure 3.5: Conversion from weight maximization to cost minimization problem

dummy Chinese tokens such that $m_1 = n_1 + n_k$. Dummy edges with weight zero are then added for connecting each dummy vertex to each vertex on the other side. The next transformation is to convert the weight of each edge $\mu(e_i, c_j)$ to the cost $\Phi - \mu(e_i, c_j)$, where $\Phi = \text{Max}_{(e_i,c_j)\in L}\mu(e_i, c_j)$. Figure 3.5 shows a sample conversion. The maximization problem is now turned to a minimization problem. We adopt the Hungarian search algorithm [15] which can solve this minimization problem efficiently. The Hungarian algorithm generates all independent sets of the matrix and computes the total costs of each assignment. The optimal solution can be obtained with polynomial time complexity.

The optimal solution of the minimum cost assignment problem can be easily converted to the optimal solution of the maximization of the total weight Ω presented in Equation 3.1. The total weight is then normalized by

the smaller number of tokens between the English and Chinese entities (i.e., $\Omega/\text{Min}(m, n)$).

Chapter 4

Phonetic Matching Model

The phonetic matching model aims at determining the similarity of two terms or word segments based on pronunciation. The first step is to generate a phonetic representation for each term. Then the similarity calculation is performed using a modified longest common subsequence algorithm.

4.1 Generating Phonetic Representation for English

4.1.1 Phoneme Generation

English terms will be processed by a phonetic generation procedure whose purpose is to generate the phonetic representation. One challenge is to han-

dle unseen English terms. We adopt the phonetic representation used in PRONLEX, a lexicon resource provided by LDC. Appendix A shows the pronunciation symbols used by the PRONLEX lexicon. Table 4.1 depicts some sample English words and the corresponding English phonetic representation.

Word	English phonetic representation
heat	hit
could	kUd
father	faDR
thin	DIs
shine	SYn
pleasure	plEZR
hang	h@G

Table 4.1: English words and the corresponding English phonetic representation in PRONLEX pronunciation symbols

An English word is first split into individual letters. Then each letter is assigned with a pronunciation tag. Some examples of English letters and the corresponding tags are shown in Table 4.2. The tag "end" in the table denotes no sound. After each letter of the word is tagged by its most likely tag according to the letter-to-phoneme lexicon, a set of transformation rules

Word	English letters with pronunciation tags
heat	h/E e/i a/end t/t
could	c/k o/U u/end l/end d/d
father	f/f a/a t/D h/end e/R r/end
thin	t/T h/end i/I n/n
shine	s/S h/end i/Y n/n e/end
pleasure	p/p l/l e/E a/end s/Z u/R r/end e/end
hang	h/h a/@ n/G g/end

Table 4.2: English words and the corresponding letters with pronunciation tags

is applied to correct some of the tags of the unseen word. The letter-to-phoneme lexicon and the transformation rules are learned from a training process using a set of training data. Details of the training process will be described in Section 4.1.2.

After generating the pronunciation tags, phonetic representation of the unseen word is obtained by grouping the pronunciation tags according to the order of letters in the unseen word. For example, the term "Blair" will be tagged as "B/b, l/l, a/e, i/end, r/r" and the result for the phonetic representation is "bler". Table 4.4 shows some tagged letters of unseen words and their phonetic representations.

The next step is to segment the representation into basic phoneme units to facilitate phonetic matching. The segmentation is determined based on the consonants depicted in Table 4.3. The phonetic representation is scanned from the beginning. If a consonant is found, then the representation will be split before it. Normally, a basic phoneme unit is composed of a consonant followed by a vowel. Sometimes it can be a single consonant or a single vowel. For example, the English phonetic representation of the word "England" is "IGglxd". After breaking down into phoneme units, the phonetic representation becomes "I G g lx d". Table 4.5 depicts the final phonetic representation in basic phoneme units of the English word. In our current model, there are 441 basic English phoneme units in total.

w, r, l, m, n, p, b, t, d, k, g, C, J, f, v, T, D, z, S, s, Z, h, y, N, G, H

Table 4.3: Consonants for determining the segmentation of the English phonetic representation into phoneme units

4.1.2 Training the Tagging Lexicon and Transformation Rules

We employ the transformation-based error-driven learning method to conduct the training of letter-to-phoneme process [3]. This learning method has

Unseen word	Tagged letters	PRONLEX symbols
Blair	b/b l/l a/e i/end r/r	bler
Nainobi	n/n a/e i/end n/n o/o b/b i/i	nenobi
Aghazade	a/@ g/g h/end a/x z/z a/e d/d e/end	@gxzed
Koppel	k/k o/a p/p p/end e/x l/l	kapxl
Bush	b/b u/U s/S h/end	bUS
Baghdad	b/b a/@ g/end h/end d/d a/@ d/d	b@d@d

Table 4.4: Tagged letters of unseen English words and their phonetic representations predicted by the tagger

Unseen word	basic phoneme units
Blair	b le r
Nainobi	ne no bi
Aghazade	@ gx ze d
Koppel	ka px l
Bush	bU S
Baghdad	b@ g d@ d

Table 4.5: Basic phoneme units of unseen English words

been applied to train part-of-speech tags. Here, we make use of this method for training the pronunciation tags. The training data was collected from PRONLEX, which contains 90,694 English words and their corresponding English phonetic representation.

A preprocessing task is needed for generating suitable training data from the PRONLEX lexicon. Each entry in the PRONLEX lexicon corresponds to an English word and its phonetic representation. The objective of the proprocessing is to align each letter in the word to a particular pronunciation tag (possibly dummy). We develop an alignment algorithm to achieve this objective. The idea of the alignment algorithm is to split words to individual letters and then assign each letter with a tag. Tagged letter is of the form "letter/tag" as shown in Table 4.1. There are 42 pronunciation symbols in total as shown in Appendix A. We manually prepare a set of letters associated with each pronunciation symbol. The set of letters is likely to be tagged as the corresponding pronunciation symbol. For example, the letters 'e', 'y', 'i' and 'z' are likely to be tagged as the pronunciation symbol 'i'. Basically, one pronunciation symbol is matched to one letter. However, there are some exceptional cases, where we would match a group of pronunciation symbols to a letter such as "b/by" and "s/zx". For the letters that are not matched with any pronunciation symbols, we would assign it with the tag "end", which indicates that it is a dummy alignment.

After the training data is prepared, the transformation-based error-driven learning method is applied. This learning method first uses statistical techniques to extract information from the training data. The learning algorithm will automatically learn a set of contextual transformation rules. At the same time, it will generate a letter-to-phoneme tagging lexicon, which contains the most likely tags of letters. It will be used for initial tagging. Then the transformation rules will be applied to reduce the errors that would be introduced by statistical mistakes of the initial tagging. Details of the learning procedures will be explained below.

Figure 4.1 depicts the modules of this transformation-based learning method. In the first stage of the learning process, a word that is not annotated is passed through an initial-state annotator. Every letter of the word is assigned with its most likely tag in isolation. The most likely tag of each English letter is depicted in Appendix B. In the second stage, the learning process starts. The learner generates a rule in each iteration according to the triggers of the contextual templates. The generated rule is used to update the tags of the annotated word. Then the annotated word is compared with the training data and see if there is any improvement in the tagging accuracy. If the number of tagging errors is reduced by applying this rule, then the rule will be recorded down as one of the contextual transformation rules and it will be used to tag the annotated word for the next iteration. Table 4.6 de-

picts some samples of the generated rules. The iteration carries on until the degree of improvement becomes insignificant. A letter-to-phoneme lexicon will also be collected for initial tagging through this learning process.

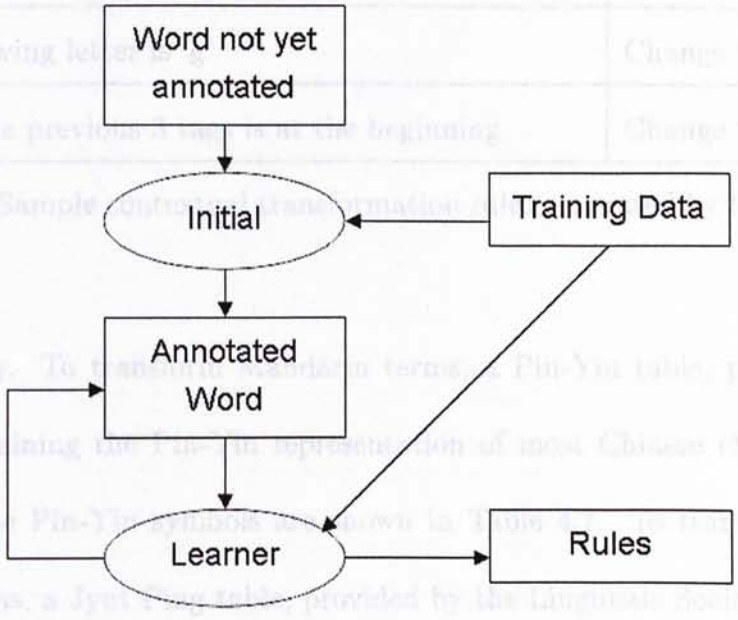

Figure 4.1: Transformation-based error-driven learning module

4.2 Generating Phonetic Representation for Chinese

Two popular spoken Chinese dialects are Mandarin and Cantonese. Chinese terms pronounced in Mandarin and Cantonese are converted into pronunciation representation by using Pin-Yin symbols and Jyut Ping symbols

Rule condition	Rule action
If the current letter is 'e' and the following letter is 'r'	Change 'end' to 'R'
If one of the previous 2 tags is 'R'	Change 'r' to 'end'
If the following letter is 'g'	Change 'n' to 'G'
If one of the previous 3 tags is at the beginning	Change 'end' to 'E'

Table 4.6: Sample contextual transformation rules generated by the learning method

respectively. To transform Mandarin terms, a Pin-Yin table, provided by LDC, containing the Pin-Yin representation of most Chinese characters is used. Some Pin-Yin symbols are shown in Table 4.7. To transform Cantonese terms, a Jyut Ping table, provided by the Linguistic Society of Hong Kong (LSHK)[1], is used. Some Jyut Ping symbols are shown in Table 4.8. After obtaining the symbol from the table, the tone is ignored and it is broken into basic phoneme units. Normally a basic phoneme unit consists of a consonant followed by a vowel similar to English. If there is no consonant-vowel pattern, we extract the consonant. If there is no consonant, the vowel will be extracted as a phoneme unit. For example, the Chinese term "貝克漢姆" pronounced in Mandarin will be transformed to basic phoneme unit sequence "bei ke han mu". The Chinese term "碧咸" pronounced in Can-

[1]The URL of LSHK is http://cpct92.cityu.edu.hk/lshk.

tonese will be transformed to basic phoneme unit sequence "bik haam". In

Mandarin character	Pin-Yin symbol
唉	ai3
八	ba1
猜	ca1
當	dang1
耳	er3
方	fang1
港	gang3
海	hai3

Table 4.7: Sample Mandarin characters with their corresponding Pin-Yin symbols

our current model, there are 791 and 1,139 Mandarin and Cantonese basic phoneme units in total respectively.

4.3 Phonetic Matching Algorithm

Given an English term and a Chinese term both represented as a sequence of basic phoneme units, we wish to calculate a similarity value which indicates how similar their pronunciations are. To achieve this, we first prepare a

Cantonese character	Jyut-Ping symbol
唉	aai1
爸	baa1
差	caa1
得	dak1
飯	faan6
今	gam1
下	haa6
音	jam1

Table 4.8: Sample Cantonese characters with their corresponding Jyut-Ping symbols

phoneme pronunciation similarity (PPS) table capturing the pronunciation similarity value between each possible English-Chinese phoneme unit pair. There are two such tables, one for English-Mandarin and the second one for English-Cantonese matching. Two different PPS tables are needed due to different pronunciations of the two dialects. The range of the pronunciation similarity values is between 0 and 1, which indicates how similar the two basic phoneme units of different languages pronounce.

We have prepared both PPS tables with pronunciation similarity values assigned manually. The complete English-Mandarin and English-Cantonese PPS tables contain 348,831 and 502,299 entries respectively. The pronunciation similarity values of most entries are zero due to completely dissimilar pronunciations. We only retain those entries which exhibit certain similarity resulting in significantly less entries. In particular, the number of entries for English-Mandarin and English-Cantonese PPS tables are 35,077 and 39,981 respectively. Table 4.9 depicts some sample entries in the manually prepared English-Mandarin PPS table.

For English-Mandarin PPS table, we also investigate several learning algorithms which can determine the similarity values automatically using a set of training data. The details of the learning algorithms are described in Chapter 5.

Suppose an English term, A, is represented by basic phoneme unit se-

English-Mandarin phoneme unit pair	Pronunciation similarity value
gi - gian	1.0
wa - hiao	0.9
HE - wiang	0.8
ZU - xu	0.8
a - ou	0.5

Table 4.9: Sample entries in English-Mandarin phoneme pronunciation similarity (PPS) table with values assigned manually

quence $<a_1, \ldots, a_{m_a}>$. A Chinese term, B, is represented by basic phoneme unit sequence $<b_1, \ldots, b_{m_b}>$. The objective of the phonetic matching model is to compute the longest matched subsequence between two phoneme sequences. The mapping must be in the same order, but not necessarily consecutive. This problem can be formulated as finding longest common subsequence (LCS). Dynamic programming can be employed to find the optimal solution for LCS efficiently. However, the basic LCS algorithm only considers the presence or absence of mapping between a_i and b_j. In contrast, in our phonetic matching problem, the matching similarity can take any value between 0 and 1. Therefore, we modify the standard dynamic programming to accept real-valued matching similarity. Let $S_{i,j}$ be the similarity score of the optimal longest common subsequence for the sequences $<a_1, \ldots a_i>$ and

$<b_1, \ldots, b_j>$. The corresponding recursive formula is depicted as follows:

$$S_{i,j} = \begin{cases} 0 & \text{when } i = 0 \text{ or } j = 0, \\ \text{Max}(S_{i-1,j-1} + V_{a_i,b_j}, S_{i,j-1}, S_{i-1,j}) & \text{otherwise} \end{cases} \quad (4.1)$$

where V_{a_i,b_j} represents the pronunciation similarity value of phoneme unit pair involving a_i and b_j. This similarity value should be found in the PPS table. S_{m_a,m_b} becomes the matching score of the intended optimal solution. This score is then normalized by the maximum length of the two sequences.

In the following, we further demonstrate how the modified longest common subsequence is used for calculating the phonetic similarity. Table 4.10 shows the calculation of the similarity value between "Beckham" and "貝克漢姆". The similarity score is in the right bottom corner. The score is normalized by the maximum length of the two phoneme sequence, i.e. 4, resulting in the finalized similarity of 0.75. Table 4.11 shows the similarity calculation of another name pair "Beckham" vs "漢武". The similarity score is normalized by the maximum length of 3, resulting in the final similarity score of 0.5. The result shows that "貝克漢姆" and "Beckham" matches better than "漢武" with "Beckham".

			j				
			0	1	2	3	4
				bei	ke	han	mu
	0		0.0	0.0	0.0	0.0	0.0
	1	bE	0.0	1.0	1.0	1.0	1.0
i	2	kx	0.0	1.0	2.0	2.0	2.0
	3	m	0.0	1.0	2.0	2.0	3.0

Table 4.10: "Beckham" vs "貝克漢姆" similarity calculation from English-Mandarin PPS table

			j		
			0	1	2
				han	wu
	0		0.0	0.0	0.0
	1	bE	0.0	0.0	0.0
i	2	kx	0.0	1.0	1.0
	3	m	0.0	1.0	1.5

Table 4.11: "Beckham" vs "漢 武" similarity calculation from English-Mandarin PPS table

Chapter 5

Learning Phonetic Similarity

As mentioned above, the phonetic matching model requires a PPS table to determine the optimal mapping subsequence between two phonetic representation. A simple strategy for preparing the PPS table is to assign the elements in the table manually. However, it is not easy even for experts to assign good precise values. For English-Mandarin PPS table, we investigate several learning algorithms for obtaining the similarity values in the PPS table using a set of training data. We extracted 20,000 Chinese-English person name pairs from the Chinese-English Named Entity Corpus provided by LDC as the training data. All these name pairs are translations of each other. The names are transformed into basic phoneme units through the procedures described in Chapter 4. Let m_a and m_b be the number of phoneme units of the two given names after transformation. Consider a PPS table

V with elements $V_{i,j}$ where i and j refer to a specific English and Chinese phoneme unit respectively. Consider the k-th pair of names. Let $U_{k,i,j}$ be a binary variable indicating the presence of the phoneme unit pair involving English phoneme unit i and Chinese phoneme unit j. The similarity score of the pair of names is proportional to $\sum_{i,j} V_{i,j}U_{k,i,j}$. The goal is to obtain V such that this similarity score is as high as possible for each correct name pair while the score is low for other names which are not the translation of each other. Consider the difference of the computed similarity score Y_k and the actual one Z_k for the k-th name pair. One can attempt to minimize the sum of the square of the difference of all training name pairs. Formally, it can be expressed as:

$$\text{Minimize} \quad \Delta = \sum_k (Y_k - Z_k)^2 \tag{5.1}$$

subject to

$$Y_k = \sum_{i,j} V_{i,j}U_{k,i,j}/\text{Max}(m_a, m_b)$$

$$0 < V_{i,j} \leq 1 \quad \forall\, V_{i,j} \in V$$

This problem appears to be a standard large-scale linear regression. Our objective is to minimize Equation 5.1 based on setting the gradient equal to zero, where each component of the gradient is calculated as:

$$\frac{\partial \Delta}{\partial V_{ij}} = 2 \sum (V * U_k - Z_k) U_{k,i,j} \tag{5.2}$$

where V is a column vector formed by V_{ij} and U_k is a row vector formed by elements of $U_{k,i,j}$. The gradient can be written as a vector of all of the derivatives as shown in Equation 5.3.

$$\frac{\partial \Delta}{\partial V} = 2\sum(V * U_k - Z_k)U_k = 2U^T(UV - Z) \tag{5.3}$$

Thus, setting the gradient equal to zero, we can calculate V as depicted in Equation 5.4, Equation 5.5, and Equation 5.6.

$$2U^T(UV - Z) = 0 \tag{5.4}$$

$$U^TUV = U^TZ \tag{5.5}$$

$$V = (U^TU)^{-1}U^TZ \tag{5.6}$$

where the quantity $(U^TU)^{-1}U^T$ is known as "pseudoinverse". It can be computed whenever U^TU is invertible. Moreover, the complexity of computing the inverse is very high. In order to address this issue, we investigate three learning algorithms described below suitable for this problem setting.

5.1 The Widrow-Hoff Algorithm

The Widrow-Hoff (WH) algorithm could solve this problem in gradient descent fashion [24]. However, when the entries in the PPS table V changes, $U_{k,i,j}$ will also vary as determined by the LCS algorithm. The standard WH algorithm assumes that the variable $U_{k,i,j}$ is fixed throughout the training

process. Therefore, a modified WH algorithm similar to the one proposed in [18] is investigated.

Ideally, Z_k should be set to the actual similarity value of the k-th name pair. To reduce manual effort, we use an approximation for Z_k. Specifically, Z_k is set to 1 for positive training examples and 0 for negative examples. In each iteration, the WH algorithm works on one name pair. Each name pair is processed sequentially and the whole training set of name pairs are processed multiple times. At iteration $t+1$, suppose it is handling the k-th name pair. The LCS algorithm is applied to obtain $U_{k,i,j}$. Then, the similarity value, $V_{i,j}(t+1)$, of a particular entry in the PPS table is updated by:

$$V_{i,j}(t+1) = V_{i,j}(t) + \psi(Y_k(t) - Z_k)U_{k,i,j}(t) \tag{5.7}$$

where $Y_k(t)$ is given in Equation 5.1 for iteration t and $\psi > 0$ is the learning rate. This process continues until a terminating condition is met. A different set of name pairs called the validation set is used to implement the terminating condition. Specifically, the validation set is evaluated for every full iteration of processing all training examples one time. If the performance of the latest trained PPS table is not improved for three full iterations, the terminating condition is met.

5.2 The Exponentiated-Gradient Algorithm

The second algorithm we investigate is the exponentiated-gradient (EG) algorithm which was introduced in [14] for linear classifiers. The top level framework of EG is similar to WH in that it processes one training name pair at a time and updates the PPS table entries immediately. EG requires that the elements in V are nonnegative and sum to 1. Equivalently, V belongs to the probability simplex. V is always maintained as a probability simplex in the whole training process. When V is used for calculating the similarity between two phoneme sequences, we magnify elements in V so that the value is comparable to the original design of the PPS table. Specifically, each element in V is divided by $\text{Max}_{i,j}(V_{i,j})$. Let $V'_{i,j} = V_{i,j}/\text{Max}_{i,j}(V_{i,j})$. We define Y'_k as:

$$Y'_k = \sum_{i,j} V'_{i,j} U_{k,i,j} / \text{Max}(m_a, m_b) \tag{5.8}$$

The updating formula is given by:

$$V_{i,j}(t+1) = V_{i,j}(t) \exp(\kappa(Y'_k(t) - Z_k) U_{k,i,j}(t))/\Psi \tag{5.9}$$

where $\kappa > 0$ is the learning rate. Ψ is a normalization expression which is the sum of the updated $V_{i,j}$. Both WH and EG attempt to minimize the squared loss expressed in Equation 5.1. At the same time, they control the elements of the new vector V to be close to the old one. Since $U_{i,j}$ is binary,

there is a theoretical justification suggesting that EG might perform well on this learning problem.

5.3 The Genetic Algorithm

One way to view the learning problem is to formulate it as an optimization problem as follows:

$$\text{Maximize} \sum_{k} V_{i,j} U_{k,i,j} \tag{5.10}$$

$$\text{subject to} \quad 0 < V_{i,j} \leq 1 \quad \forall\, V_{i,j} \in V$$

where the summation is conducted over all positive training examples. Due to the nature of the dependency of $U_{i,j}$ on V, this optimization problem cannot be solved analytically. We investigate a genetic algorithm to solve this optimization problem. Genetic algorithms have been shown to perform well in different kinds of optimization problems. In our genetic algorithm formulation, a chromosome represents all the elements in the PPS table. Each gene in a chromosome corresponds to a particular element in the table. An initial population of chromosomes is prepared. Standard genetic operators such as crossover and mutation rates are employed.

Chapter 6

Experiments on Named Entity Matching Model

To measure the performance of the named entity matching model, two sets of experiments have been conducted. The first set of experiments is to evaluate the phonetic similarity learning. We collected 20,000 pairs of person names from the Chinese-English Named Entity Corpus provided by LDC as the training data. A validation set of 2,000 pairs of person names were collected from the same corpus for the termination condition in WH and EG algorithms. We further collected 2,000 person name pairs separated from the training and validation data to evaluate the learning performance. The second set of experiments is to evaluate the performance of the overall named entity matching model. We collected 1,000 named entities from the same

corpus for evaluation in the second set of experiments.

For evaluation purpose, all English entities and Chinese entities are regarded as two sets of entities. For each Chinese entity, a similarity score is computed between the Chinese entity and each English entity. Then, the English entities are ranked by their similarity scores in descending order. The average reciprocal rank (ARR) [23] is used to measure the performance as follows:

$$ARR = \frac{1}{N}\sum_{i=1}^{N}\frac{1}{r_i} \tag{6.1}$$

where N is the total number of unique Chinese entities; r_i is the rank of the corresponding correct English entity. The value of ARR is between 0 and 1. The higher the ARR value, the better the performance is. ARR rewards correct translations near the front of the ranked list and penalizes translations near the end of the list.

6.1 Results for Learning Phonetic Similarity

Three learning approaches, namely, WH, EG, and genetic algorithm were evaluated and compared. The initial $V_{i,j}(0)$ for both WH and EG algorithms were initialized to the elements similar to the manual PPS table. For WH algorithm, the learning rate ψ, was set to $5e10^{-5}$. For EG algorithm, the learning rate κ, was set as 0.01. For genetic algorithm, the initial populations

were initialized based on the manual PPS table. The crossover rate and the mutation rate were obtained using a validation data set via a tuning process. The best crossover rate and mutation rate were found to be 0.8 and 0.0001 respectively. For all three learning algorithms, after a PPS table has been learned, we applied the PPS table on the testing data and evaluated by the ARR score as defined in Equation 6.1.

The result is shown in Table 6.1. The performance of the manual assignment of phonetic similarity information in the PPS table is 0.780. As expected, all learning algorithms can produce a PPS table with better performance than the manual assignment. In addition, both EG and genetic algorithms perform slightly better than WH. Both WH and EG were quite efficient with running time of 10 minutes. However, genetic algorithm required considerable amount of time and it took about half an hour to run.

Training Algorithm	ARR
WH	0.863
EG	0.895
Genetic Algorithm	0.890

Table 6.1: The ARR results of different learning algorithms

6.2 Results for Named Entity Matching

A set of named entities is used to evaluate the performance of the named entity matching model. We also conducted the same experiment on the pure phonetic model and the pure semantic model for comparison. The pure phonetic model only makes use of phonetic information without using the bilingual dictionary. It is implemented by restricting the named entity matching model described in Chapter 3 to only using phonetic evidences. The pure semantic model only considers translations found in the dictionary without using phonetic information. It is implemented by restricting the named entity matching model to only using the dictionary.

Model	ARR
Named entity matching model	0.802
Pure semantic model	0.767
Pure phonetic model	0.423

Table 6.2: The ARR results of the named entity matching, pure semantic, and pure phonetic models

The ARR of named entity matching, pure semantic, and pure phonetic models are 0.802, 0.767, and 0.423 respectively as shown in Table 6.2. The result of the pure phonetic model has demonstrated that the phonetic model alone is not a good model to handle named entities. This is because most

of the names consist of multiple terms, with part of them translated based on meaning rather than pronunciation. Moreover, the terms are not translated according to the original order. Thus, when we simply transform the names into phonetic representation and conduct the matching, the effectiveness is limited. The result of the pure semantic model is better than the pure phonetic model indicating that many named entities possess a considerable amount of semantic evidence. The performance of our proposed named entity matching model is the best among the other models. It clearly demonstrates that considering both semantic and phonetic information is an advantage.

Chapter 7

Mining New Entity Translations from News

A useful application of our named entity matching model is to discover new, unseen named entity translations from online daily Web news. We develop a mining approach that first automatically harvests bilingual comparable news clusters by analyzing the content of the news stories. Unsupervised learning technique using a bilingual dictionary is employed to detect comparable news clusters. New named entity translations not found in the existing bilingual dictionary are then discovered by applying our named entity matching model.

Figure 7.1 depicts the components of this mining approach. Online daily Web news stories from different agencies or sources are downloaded and collected. Currently, our system fetches online English and Chinese news from

seven sources as shown in Table 7.1. The first three sources offer English news while the remaining ones provide Chinese news.

http://www.cnn.com	
Source: Cable News Network (CNN)	Lang: English
http://www.un.org/av/radio/news/latenews.htm	
Source: United Nations Radio (UNR)	Lang: English
http://www.rthk.org.hk/rthk/news/englishnews	
Source: Radio Television Hong Kong (RTHK)	Lang: English
http://www1.chinadaily.com.cn/gb/worldinfo/foreign.html	
Source: China Daily (CND)	Lang: Chinese
http://news.yam.com/afp/international	
Source: Agence France-Presse (AFP)	Lang: Chinese
http://www.zaobao.com	
Source: Zao Bao (ZAO)	Lang: Chinese
http://www.rthk.org.hk/rthk/news/expressnews	
Source: Radio Television Hong Kong (RTHK)	Lang: Chinese

Table 7.1: Online news sources

Metadata information for each news story is automatically generated. The metadata information includes story ID, source, release time, language, and named entity information. Named entities refer to people names, place

names, and organization names. A sample of metadata is shown in Table 7.2. Two automatic named entity extraction methods are developed to extract the named entities for each story, one for English and one for Chinese. Some attribute information about named entities are also detected at the same time.

Story ID	20031122_003_CNN.e
Source	CNN
Release Time	2003/11/20
Language	English
People Name	Kerem Yilmazr (origin=Foreign) Jack Straw (origin=Foreign)
Place Name	United States Turkey
Organization Name	National Security Council

Table 7.2: A sample metadata

A context-based gloss translation method is developed to conduct gloss translation from the Chinese content terms of Chinese stories into English terms so that we have a uniform representation for each story for event discovery purpose. The content terms of each news story are further processed by information retrieval techniques and the news story is represented by a

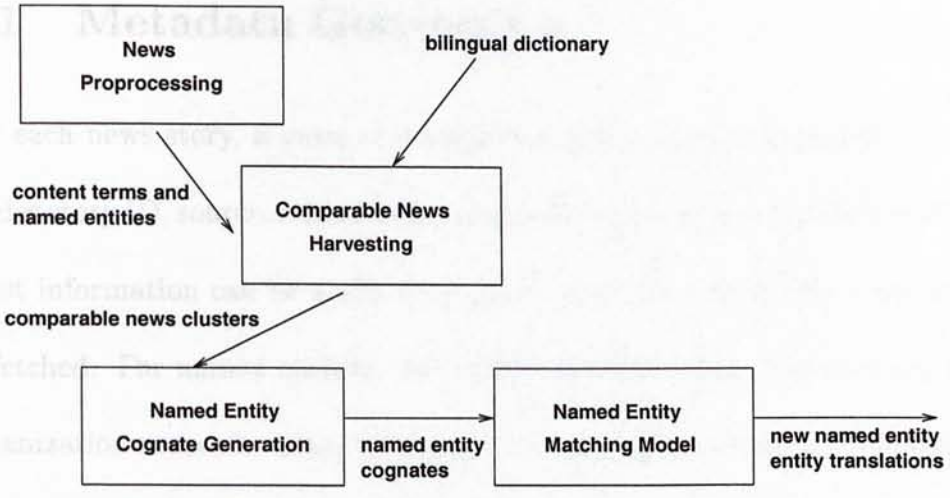

Figure 7.1: Components of the new, unseen named entity translation mining approach

four-dimensional vector with appropriate weights. Unsupervised learning is employed to discover new events and track previously discovered events. A sample of a discovered event is shown in Table 7.7. The stories in different languages in a particular event can be treated as comparable news. Named entity candidates can be generated based on the usage of named entity of the comparable news reporting the same event. A named entity matching algorithm based on phonetic and context clues is developed and new unseen name translations can be discovered. A sample of some discovered new name translations is given in Table 8.1.

7.1 Metadata Generation

For each news story, a piece of metadata is automatically generated. It includes story ID, source, release time, language, and named entity information. Most information can be easily determined from the source where the news is fetched. For named entities, we extract people names, place names, and organization names by using automated named entity extraction techniques. To extract Chinese named entities, we use a probabilistic rule-based extraction approach. We first extract all possible named entity cognates according to a set of rules and assign each named entity cognate a probability value [4]. The probability value of each cognate represents its likelihood to be a certain kind of named entity. Then we employ a pattern analysis technique to select the most likely named entities [25]. To extract English named entities, we make use of a statistical approach based on a variant of the standard hidden Markov model [2].

We also detect attribute information for each people name including the origin of the name, the surname, and the given name. For the people names in Chinese, we determine three kinds of origin, namely, Chinese, Japanese, and Foreign. Since the rules we used to extract named entities from Chinese news contain the constraints of origin, surname, and given name information. These attribute information can be determined at the same time when the

people names in Chinese are extracted. For the people names in English, we further divide the Chinese origin into Mandarin and Cantonese. We make use of several resources of surname lists containing a large number of surnames. The resource containing Japanese surnames in English is obtained from JMnedict (Japanese Multilingual Named Entity Dictionary)[1]. The resource of Mandarin and Cantonese surnames in English are both obtained from Wikipedia[2]. By comparing each token of the people name with the entries of surname lists, we can identify the origin of the people name. For those people names that have a surname entry in both Mandarin and Cantonese lists, we will further analyze the given name with the help of the PinYin list. We also perform name canonicalization on people names in English with origin of Mandarin, Cantonese, and Japanese. The name canonicalization process refers to re-arranging the words in the name so that the given name will be placed following the surname.

[1]JMnedict can be obtained from the Monash Nihongo ftp Archive (http://ftp.cc.monash.edu.au/pub/nihongo/).

[2]The URL of Wikipedia is http://en2.wikipedia.org/wiki/Chinese_family_name.

7.2 Discovering Comparable News Cluster

7.2.1 News Preprocessing

The content of news is processed using information retrieval techniques. For English news, stemming and stop-word removal are applied. For Chinese news, word segmentation and stop-word removal are applied. The word segmentation is performed based on maximizing the segmented token probability via dynamic programming. Let S be a news story. The story representation comprises of four components, namely, people name component $R_p(S)$, place name component $R_l(S)$, organization name component $R_o(S)$, and content term component $R_c(S)$. Each component is represented by a set of weighted terms shown as follows:

$$R_p(S) = (w(S, p_1), w(S, p_2), \dots)$$

$$R_l(S) = (w(S, l_1), w(S, l_2), \dots)$$

$$R_o(S) = (w(S, o_1), w(S, o_2), \dots)$$

$$R_c(S) = (w(S, c_1), w(S, c_2), \dots)$$

where $w(S, p_i)$, $w(S, l_i)$, $w(S, o_i)$, and $w(S, c_i)$ represent the weights of the corresponding people name p_i, place name l_i, organization name o_i, and content term c_i in the news story S respectively. The weight of each term is determined by several factors. One factor is the term frequency defined

as the number of occurrence of a term in the story. The term frequency is also adjusted by the relative location of the term in the content of the story. Another factor is the incremental document frequency.

7.2.2 Gloss Translation

Purpose and Principle

In dealing with cross-language comparison, we conduct context-based gloss translation on Chinese terms. Chinese story representation will be translated into English story representation so that we can perform unsupervised learning on an uniform language representation. Gloss translation approach is adopted instead of full-fledged machine translation since our objective is to translate Chinese terms adequately for comparable news clusters discovery purpose.

For each Chinese term, we will look up a bilingual lexicon for the English translation. The translated English terms will then replace the original Chinese terms to represent the story. Typically, a set of translated English terms can be found. Some translated terms are inappropriate to represent the meaning of the original term. Thus, term weights are adjusted so that the more likely translated terms will receive more emphasis. We design a context-based model to perform term disambiguation inspired by [9].

This model is based on the principle that correct translations tend to co-occur together and incorrect ones do not. It disambiguates English term translations by making use of the mutual information between terms.

Several sets of experiments on translating Chinese and Arabic documents are also conducted in Chapter 9 to compare the context-based model with other translation schemes.

Formulation

In the following, we will explain the formulation of the context-based translation model in detail. Suppose a document contains a number of Chinese sentences. In a certain sentence, there is a sequence of Chinese terms $<c_1, \ldots, c_n>$. For each Chinese term c_i, we first retrieve a set of translations $T_i = \{t_{i1}, \ldots, t_{im}\}$ found in the bilingual dictionary. Within a certain predefined window size σ, we then compute the cohesion for each English translation t_{ij} with each of the neighouring set of translation T_i. Window size refers to the number of neighbor terms on the right and left hand side of c_i involved in the co-occurrence calculations. For example, suppose we have a sequence of Chinese terms $<c_1, c_2, \ldots, c_6>$. If the window size σ is set to 2, then the cohesion of translations in c_4 will involve two Chinese terms before it and after it, i.e., c_2, c_3, c_5, and c_6. The outline of the greedy algorithm applied on our context-based translation model is shown in Figure 7.2.

```
1   For each Chinese term c_i (i=1 to n)
2       Retrieve the translations T_i = {t_i1, ... , t_im};
3   For each Chinese term c_i (i=1 to n)
4       For each translation t_ij (j=1 to m)
          /* σ is the window size */
5           Compute Cohesion(t_ij, T_k) where k=i-σ to i+σ and i ≠ k;
6           Compute the sum of Cohesion(t_ij, T_k) as score(t_ij);
```

Figure 7.2: Outline of the context-based translation algorithm

For each English translation, its total cohesion score is the sum of the cohesions with all its target words within the predefined window. The higher total cohesion score, the higher the term weight it will receive. Finally, this score is normalized by sum normalization. In order to adjust the quality and quantity of the translation output, we introduced a translation weight cutting threshold μ in our model. Only those translations with score greater than this cutting threshold will be returned as output.

The cohesion between term x and a set T of neighbor terms is computed as follows:

$$Cohesion(x, T) = \sum_{y \in T} SIM(x, y) \tag{7.1}$$

In the above equation, we define a particular English translations of c_i to be x and y is one of the English translations of c_j where c_j is a neighbor of c_i, which is within the predefined window.

The term similarity calculation involves two components, namely, mutual

information and distance factor as shown in Equation 7.2.

$$SIM(x,y) = MI(x,y) * D(x,y) \tag{7.2}$$

In the above equation, distance factor $D(x,y)$ is introduced. As closer terms are more likely to have a stronger relationship and thus more similar. The formulation is shown in Equation 7.3. The factor decreases proportionally when the distance between term x and y increases.

$$D(x,y) = 1 - \alpha * (Dis(x,y) - 1) \tag{7.3}$$

where α is the decay rate and *Dis(x,y)* is the distance between term x and y.

The mutual information MI mentioned in Equation 7.2 is estimated according to their co-occurrence frequency within a certain window size. It is defined as follows:

$$MI(x,y) = P(x,y) * log(\frac{P(x,y)}{P(x) * P(y)} + 1) \tag{7.4}$$

where

$$P(x,y) = \frac{C(x,y)}{\sum_{x',y'} C(x',y')}, P(x) = \frac{C(x)}{\sum_{x'} C(x')} \tag{7.5}$$

C(x,y) is the number of co-occurrences of terms x and y within a lexical unit (e.g. a sentence) of an English collection of documents. *C(x)* is the number of occurrences of term x in the same collection.

The following example illustrates the context-based gloss translation model on a Chinese sentence "中國 學術 研究 穩步上揚 階段". Suppose the

window size σ is set to 2 and α is set to 0.2. Consider the term disambiguation task for the English translations of the term "研究". Table 7.3 depicts the translations for the Chinese terms in the sentence with their corresponding term frequencies (i.e. $C(x)$). The co-occurrence statistics be-

Chinese Term	Translation $x_1(c(x_1))$, Translation $x_2(c(x_2))$
中國	Cathay(3), China(9)
學術	academic(8)
研究	analyze(15), research(4)
穩步上揚	N/A
階段	stage(2)

Table 7.3: Translations and term frequencies for the Chinese terms

tween the English translations is shown in Table 7.4. The probability that

	Cathay	China	academic	stage
analyze	2	5	6	0
research	1	1	3	1

Table 7.4: Co-occurrence statistics $C(x, y)$ between the corresponding English translations

"analyze" and "Cathay" co-occur in the same sentence in the training corpus is computed using Equation 7.5. Equation 7.6 illustrates the actual

computation. The probabilities that "analyze" and "Cathay" appear in the corpus are illustrated in Equation 7.7 and Equation 7.8 respectively. After this, we can compute the mutual information (MI) between "analyze" and "Cathay" as shown in Equation 7.9. The calculation of distance factor is depicted in Equation 7.10. The term similarity is computed by multiplying MI and $D(x, y)$, which is depicted in Equation 7.11. Similarly, we got all the $SIM(x, y)$ of "analyze" with other terms as shown in Table 7.5. The cohesions between "analyze" and the neighboring Chinese terms are shown in Table 7.6. By summing up the cohesions within the predefined window, the preliminary translation score is computed in Equation 7.12. Similarly, $score(research)$=1.79. After normalization, the final translation score of "analyze" is 0.32 and that of "research" is 0.68.

$$P(analyze, Cathay) = \frac{2}{2+5+1+1} = 0.22 \tag{7.6}$$

$$P(analyze) = \frac{15}{15+4} = 0.79 \tag{7.7}$$

$$P(Cathay) = \frac{3}{3+9} = 0.25 \tag{7.8}$$

$$MI(analyze, Cathay) = 0.22 * log(\frac{0.22}{0.79 * 0.25} + 1) = 0.16 \tag{7.9}$$

$$D(analyze, Cathay) = 1 - 0.2 * (2 - 1) = 0.8 \tag{7.10}$$

$$SIM(analyze, Cathay) = 0.16 * 0.8 = 0.13 \quad (7.11)$$

	Cathay	China	academic	stage
analyze	0.13	0.3	0.41	0.0

Table 7.5: Term similarity scores $SIM(x, y)$ between "analyze" and "研究"'s neighboring terms' English translations

	中國	學術	穩步上揚	階段
analyze	0.13+0.3=0.43	0.41	0.0	0.0

Table 7.6: The cohesions between "analyze" and "研究"'s neighboring Chinese terms

$$score(analyze) = 0.43 + 0.41 = 0.84 \quad (7.12)$$

Co-occurrence Training

In our context-based gloss translation model, we need to collect the term frequencies and co-occurrence frequencies for the calculation. This information is trained from the Topic Detection and Tracking 4 (TDT-4) English news corpus. This corpus contains news from various news agencies including Associated Press Worldstream Service and New York Times. We collected the

news items from January 2000 to January 2001, with 600,000 sentences in total for training. We computed all term frequencies of all terms as well as co-occurrence frequencies for every term pair within a sentence.

7.2.3 Comparable News Cluster Discovery

A comparable news cluster is also represented by a four-dimensional vector similar to the story representation. Incremental nearest neighbor clustering is used for processing the stories. An incoming story is compared to all existing comparable news clusters according to a similarity measure. The closest comparable news cluster can be determined. If the final normalized similarity δ_f of the story to the closest comparable news cluster is larger than a user defined threshold θ, the story will be inserted into the comparable news cluster. In this case, the comparable news cluster representation will be updated. Otherwise, the story will form a new cluster on its own representing a new comparable news cluster. By changing this threshold θ, we can adjust the granularity of comparable news cluster.

We use a kind of cosine similarity measure to compute the similarity between a comparable news cluster and a story. For each component of the story and comparable news cluster representation, a similarity score is calculated by processing the weights of the terms in the component. For instance,

we compute the similarity score δ_p between the people name component of the story S and the comparable news cluster E by the following formula:

$$\delta_p = \frac{\sum_k \sum_l a_{kl}}{\sqrt{\sum_l w(S,p_l)^2 \sum_k w(E,p_k)^2}} \tag{7.13}$$

$$a_{kl} = \begin{cases} w(S,p_l)w(E,p_k) & \text{when } p_l = p_k \\ 0 & \text{when } p_l \neq p_k \end{cases} \tag{7.14}$$

where $w(S,p)$ is the weight of the people name p in the story S and $w(E,p)$ represents the weight of the people name p in the comparable news cluster E. We can compute the similarity score δ_l for the place name component; δ_o for the organization name component; and δ_c for the content term component in a similar manner. The final similarity δ_f, is a weighted sum of these similarity scores:

$$\delta_f = \delta_p\Lambda_p + \delta_l\Lambda_l + \delta_o\Lambda_o + \delta_c(1 - \Lambda_p - \Lambda_l - \Lambda_o) \tag{7.15}$$

$$\Lambda_p, \Lambda_l, \Lambda_o \geq 0$$

$$\Lambda_p + \Lambda_l + \Lambda_o \leq 1$$

where Λ_p, Λ_l, and Λ_o are the corresponding component weights. By adjusting these component weights, we can specify the relative contribution of each component to the final similarity.

It is common that stories reporting the same comparable news cluster are released in a short period of time (e.g., several days). We introduce a

time adjustment factor T to cope with this phenomenon as shown in Equation 7.16.

$$T = \begin{cases} 1.0 + \frac{|d_s - d_a|}{10}(L_p - 1) & \text{when } |d_s - d_a| < 10 \\ 0 & \text{when } |d_s - d_a| \geq 10 \end{cases} \tag{7.16}$$

where d_s is the release date of the story; d_a is calculated by the average release days of all stories belonging to the event; L_p is the time adjustment threshold. We assume that when a news story happens more than ten days away from an event, it is not likely that this story is related to the comparable news cluster. A sample comparable news cluster is given in Table 7.7.

Title	Source	Release date
Turkey buries latest bomb victims	CNN	2003 Nov 22
Arrests made in Turkey blasts	CNN	2003 Nov 21
Arrests over Istanbul bombings	RTHK	2003 Nov 21
英駐土耳其總領事在爆炸中喪生	AFP	2003 Nov 21
伊斯坦布爾發生至少兩次劇烈爆炸	CND	2003 Nov 20

Table 7.7: A sample comparable news cluster

7.3 Named Entity Cognate Generation

Typically a comparable news cluster contains several news stories. A simple filtering process is designed to select those comparable news clusters con-

taining both English and Chinese news. We generate named entity cognates from these comparable news stories. Specifically, we extract people names, place names, and organization names from the stories in a particular cluster resulting in a pair of English and Chinese named entity cognates. There is a cognate weight associated with each name in the named entity cognates. The cognate weight attempts to reflect the importance of the name in the corresponding comparable news cluster. A sample of a pair of named entity cognates for a particular cluster is shown in Table 7.8.

English cognate	(Abdullah Gul 0.511), (Istanbul 0.295), (Abu Musab 0.097), $\cdots$
Chinese cognate	(阿卜杜卡迪爾 0.121), (蕭特 0.171) (默瑟 0.184), $\cdots$

Table 7.8: A sample pair of named entity cognates

The cognate weight is calculated separately for people and place names, as well as for English and Chinese names. For example, the cognate weight of each English people name E_i in a particular comparable news is calculated by the following formula:

$$u(E_i) = \sum_{S_j} w(S_j, E_i) \tag{7.17}$$

where $u(E_i)$ is the cognate weight of people name E_i. The final cognate weight $u(E_i)$ will be adjusted by the maximum normalization among the

cognate weights in the same comparable news cluster. The cognate weight for each place name in the English name cognate is calculated in a similar way. Hence, we can generate a set of English name cognates by collecting the names from the English news of the comparable news and calculating their corresponding weights. The Chinese name cognates are obtained in a similar manner. As a result, a pair of bilingual named entity cognates is generated from each cluster as shown in Table 7.8.

7.4 Entity Matching

7.4.1 Matching Algorithm

As mentioned before, our model considers phonetic similarity, cognate weighting and entity origin. Cognate weighting is determined from the analysis of the content weight of the name in stories belonging to the same comparable news cluster. The weighting can be obtained from the cognate generation process mentioned in Section 7.3. The cognate weighting indicates the relative importance of the name in a particular comparable news. The higher the weighting, the more important the name is. If both of the English and Chinese names are of relatively high weightings in a particular comparable news, they are more likely to be a matched pair. On the contrary, if both

of them are of low weightings, they are less likely to be matched. In cases just one of them is of high weighting while the other one is not, they are also unlikely to be matched. The formula for measuring the cognate weighting similarity score $S_w(E, C)$, of an English name E and a Chinese name C is defined as follows:

$$S_w(E, C) = \min(u(E), u(C)) \tag{7.18}$$

where $u(E)$ and $u(C)$ are the candidate weighting of E and that of C in a particular event respectively. Only those names which are both with high weightings will result in high similarity score.

Entity origin information is obtained from the metadata generation component. The origin information can be utilized to predict the likeliness of whether an English name should be matched with a particular Chinese name. Generally, if both of the English and Chinese names come from the same origin, they are more likely to be a match. Let $S_o(E, C)$ denote the similarity score due to the origin information. If both of the English and Chinese names come from Chinese origin, $S_o(E, C)$ is assigned to 1.0. The same scheme is applied to the names which are both from Japanese origin. However, if both of the names are names of foreign origin, $S_o(E, C)$ is set to 0.8. If either the English or the Chinese name comes from foreign origin, $S_o(E, C)$ is set to 0.6 because there may still be some chance for a name from foreign origin

belonging to Chinese or Japanese category in case the existing name lexicons have not covered that entry. For the remaining cases such as an English name of Chinese origin with a Chinese name of Japanese origin, $S_o(E, C)$ is set to 0 as it is obvious that they are very unlikely to be a matched pair.

The final similarity score $S_f(E, C)$, is given as follows:

$$S_f(E, C) = S_p(E, C)\alpha_p + S_o(E, C)\alpha_o + S_w(E, C)(1 - \alpha_p - \alpha_o) \quad (7.19)$$

where α_p and α_o are the two parameters controlling the relative contribution of phonetic similarity, cognate weighting, and entity origin.

7.4.2 Matching Result Production

The system attempts to discover new name translations on a daily basis. At a certain day, the system collects possible name discovery from each cognate. Consider a cognate, the final similarity score $S_f(E, C)$ of each pair of the names is evaluated. All the possible pairs under the same Chinese name will be grouped together. The corresponding English names will be output according to the final similarity scores with the Chinese name and sorted in descending order. Those English names with the similarity score greater than a threshold ϕ, will be returned. Those names will be stored and users can retrieve them for other future uses.

Chapter 8

Experiments on Mining New Translations

We have conducted an experiment to evaluate the named entity discovery performance. News articles from 20 November 2003 to 20 December 2003 are fetched from the sources listed in Table 7.1. There are 1,599 English news and 2,476 Chinese news in total. The comparable news cluster discovery process was conducted incrementally in batch in each day. Each batch contains news from four consecutive days, resulting in 28 batches in total. Comparable news clusters are generated for each batch. A number of entity translations are discovered from the clusters in each day, including some seen and unseen names. The unseen translations (not found in the bilingual dictionary) are automatically archived in our online named entity digital library so that users

can browse or retrieve them for other purposes if needed. In our experiments, the parameters α_p and α_o in Equation 7.19 were set to 0.6 and 0.2 respectively. The setting was determined by a tuning process.

Table 8.1 shows the unseen names discovered in Day 1. The output threshold ϕ mentioned in Section 7.4.2 was set to 0.5. It is interesting to see that some of the English translations are mapped with several Chinese entities due to slightly different translations from various news agencies. For example, both "佩爾利" and "帕里" are mapped with "Blair". There are in total 28 batches and 128 unseen name translations discovered. Appendix C lists the unseen named translations discovered in the remaining days. The results demonstrate that our system can successfully discover some unseen name translations not found in the existing bilingual dictionary from daily online news.

To further evaluate the quality of the mining system, we adopt ARR as the evaluation metric to measure how accurate our model can rank the correct translations for unseen names. The average ARR across all 28 days for all the named entities was 0.960. We also conducted an in-depth investigation on the performance of different kinds of entities. We found that the ARR for person names was 0.952 and the ARR for place and organization names was 0.984. As demonstrated in the ARR scores, the overall performance is very encouraging.

阿卜杜卡迪爾: Abdullah Gul, Mr Erdogan
羅伯森: Robertson, Ritz
布希: Bush
勒姆: Gul, Kerem
約翰: John
布勒: Blair, Bush
克洛: Clark, Mr Randt
庫德: Kurdish, Kimmitt, Puk, Kurdistan, Mark Kimmitt, Krivo
佩爾利: Blair, Tony Blair, Therese Munn
埃雷: Alan Greenspan, Mr Greenspan
厄姆: Munn, Therese Munn
帕里: Blair, London
伊萬諾夫: Ivanov, Mr Ivanov, Gor Ivanov

Table 8.1: Unseen name translations discovered in Day 1

Chapter 9

Experiments on Context-based Gloss Translation

In Chapter 7, we have presented the context-based gloss translation model. In this chapter, we evaluated and compared our context-based model with two other gloss translation models. Recall that the context-based model is generic enough to handle various languages. Therefore, besides the translation of Chinese documents , we also conducted translation of Arabic documents. Section 9.1 and Section 9.2 present the experimental results of gloss translation of Chinese documents and Arabic documents respectively.

9.1 Results on Chinese News Translation

The Chinese news translation evaluation is based on the TDT-3 and TDT-4 corpora from the Linguistic Data Consortium (LDC). We extracted the newswire Chinese sources (Xinhua, Zaobao, and Voice of America). The TDT-3 news are between October 1 and December 31, 1998 and the TDT-4 news are from October 1, 2000 to January 31, 2001. There are 12,341 articles from TDT-3 corpus and 25,405 articles from TDT-4 corpus. Since the machine translation version of documents is also provided by LDC, we can treat the machine translation version as the approximately correct translations which can be used for the evaluation of our gloss translation model. Specifically, we compare the translated terms from the gloss translation model with the terms from the machine translation version of the same story. The performance can be then evaluated by recall, precision, and F-measure.

We have conducted several sets of experiments on different parameter settings. The decay rate α mentioned in Section 7.2.2 was set to 0.2 as determined by the preliminary tuning experimental results.

We investigate the effect of the window size σ (defined in Chapter 7.2.2) on the gloss translation quality. We conducted the experiment for various window sizes for both TDT-3 and TDT-4 data. The translation weight cutting threshold μ (defined in Chapter 7.2.2) was set to 0.7.

Table 9.1 depicts the gloss translation performance measured by F-measure of our context-based model for different window sizes. For the results of other translation weight cutting thresholds, please refer to Appendix D and E. The results show that increasing window size has an improvement on the translation quality.

	Window Size 1	Window Size 2	Window Size 3
TDT-3	0.4073	0.4263	0.4304
TDT-4	0.4051	0.4186	0.4330

Table 9.1: The gloss translation performance measured by F-measure of our context-based model for different window sizes

We conducted another two sets of experiments to compare the performance between our context-based model and two other existing gloss translation models. One of the existing models is the equal-weighting approach. For the equal-weighting approach, equal weights are assigned to all the English translations found in the dictionary. For example, if there are four translations retrieved from the dictionary in total, then each of the English term will be assigned a weight of 0.25. The second existing gloss translation model is called the usage-factor model. Basically, the term weights of the term translations are determined by certain co-occurrence information derived from a parallel corpus. The details of the usage-factor model can be

found in [13].

We also need TDT-4 and TDT-3 data to conduct this comparative evaluation. We set the window size σ as 3 and the translation weight cutting threshold μ to 0.7 similar to the previous experiment.

The translation performance measured by F-measure of the three gloss translation models is depicted in Table 9.2. The results of other translation weight cutting thresholds can be found in Appendix D and E. The result shows that our model can effectively disambiguate English term translations.

	Context-based	Equal-weighting	Usage-factor
TDT-3	0.4304	0.2201	0.1767
TDT-4	0.4330	0.2515	0.1665

Table 9.2: The translation performance measured by F-measure of different gloss translation models

9.2 Results on Arabic News Translation

For arabic news translation, we also conducted experiments on TDT-3 and TDT-4 data. We extracted all 15,879 Arabic articles in TDT-3 corpus. It contains news from Agence France-Presse between October 1, 1998 and December 31, 1998. In TDT-4, we extracted all 41,426 newswire and broadcast

news articles from Al Hayat, An-Nahar, Agency France Press, Voice of America, and Nile TV. The articles are between October 1, 2000 and January 31, 2001. Similar to Chinese news evaluation, the machine translation version of the Arabic news is available. Hence, it can be used to conduct the evaluation of the gloss translation quality. For our context-based model, the window size σ was set to 3 and decay rate α was set to 0.2. We also conducted experiment for the equal-weighting model. The translation weight cutting threshold μ was set to 0.7 for both models.

Table 9.3 depicts the gloss translation performance measured by F-measure of our context-based model and the equal-weighting model. For the results of other translation weight cutting thresholds, please refer to Appendix F and G. The result shows that the context-based translation model also works well in conducting gloss translation for Arabic news. Our context-based model works much better than the equal-weighting model.

	Context-based	Equal-weighting
TDT-3	0.2858	0.1244
TDT-4	0.4000	0.0999

Table 9.3: The gloss translation performance measured by F-measure of our context-based model and the equal-weighting model

Chapter 10

Conclusions and Future Work

We have developed a novel named entity matching model which considers both semantic and phonetic information. We have investigated three learning algorithms on training the phonetic similarity information from training examples. Our model has been compared with the pure phonetic and pure semantic models. The experimental results show that our hybrid model can handle named entity matching in a more flexible and comprehensive way. We have also applied our named entity matching model on mining new, unseen name translations from real-world online daily Web news. Name translations not found in the dictionary can be effectively discovered from daily news.

We plan to extend our work in several directions in the future. One possible direction is to apply this model to language pairs other than English and Chinese. Another direction is to investigate the automatic utilization

of the newly discovered translations from the online Web news for improving intelligent text processing tasks such as multilingual comparable news discovery.

Bibliography

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, 1993.

[2] D. M. Bikel, R. L. Schwartz, and R. M. Weischedel. An algorithm that learns what's in a name. *Machine Learning*, 34(1-3):211–231, 1999.

[3] E. Brill. Transformation-based error-driven learning and natural language processing: A case study in part of speech tagging. *Computational Linguistics*, 21(4):543–565, 1995.

[4] H. Chen, Y. Ding, S. Tsai, and G. Bian. Description of the NTU system used for MET. In *Proceedings of 7th Message Understanding Conference*, 1998.

[5] H.-H. Chen and L.-W. Ku. "An NLP and IR Approach to Topic Detection". In *Topic Detection And Tracking: Event-based Information Organization*, pages 243–261. Kluwer Academic Publishers, 2002.

[6] P.-S. Cheung, R. Huang, and W. Lam. Financial activity mining from online multilingual news. In *Proceedings of ITCC 2004 International Conference on Information Technology: Coding and Computing*, pages 267–271, 2004.

[7] P. Fung. A statistical view on bilingual lexicon extraction: From parallel corpora to non-parallel corpora. In *Proceeding of The Association for Machine Translation in the Americas*, pages 1–17, 1998.

[8] P. Fung and L. Y. Yee. An IR approach for translating new words from nonparallel, comparable texts. In *Proceedings of the Thirty-Sixth Annual Meeting of the Association for Computational Linguisti cs and Seventeenth International Conference on Computational Linguistics*, pages 414–420, San Francisco, California, 1998.

[9] J. Gao, M. Zhou, J.-Y. Nie, H. He, and W. Chen. Resolving query translation ambiguity using a decaying co-occurrence model and syntactic dependence relations. In *Proceedings of 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 183–190, August 2002.

[10] F. Huang and S. Vogel. Improved named entity translation and bilingual named entity extraction. In *Proceedings of IEEE 4th International Conference on Multimodel Interfaces (ICMI 2002)*, pages 253–258, 2002.

[11] F. Huang, S. Vogel, and A. Waibel. Automatic extraction of named entity translingual equivalence based on multi-feature cost minimization. In *Proceedings of 41st Annual Conference of the Association for Computational Linguistics (ACL 2003), Workshop on Multilingual and Mixed-Language Named Entity Recognition*, July 2003.

[12] F. Huang, S. Vogel, and A. Waibel. Extracting named entity translingual equivalence with limited resources. *ACM Transactions on Asian Language Information Processing (TALIP)*, 2:124–129, June 2003.

[13] R. Huang, W. Lam, and Y.-Y. Law. Discovering multilingual news events and term asssociations from the web. In *Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics*, pages 226–230, July 2003.

[14] J. Kivinen and M.K. Warmuth. Exponentiated gradient versus gradient descent for linear predictions. *Information and Computation*, 132(1):1–63, 1997.

[15] H. Kuhn. The Hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, 2:83–97, 1955.

[16] W. Lam, P.-S. Cheung, and R. Huang. Mining events and new name translations from online daily news. In *Proceedings of the Joint Conference on Digital Libraries (JCDL 2004)*, 2004.

[17] W. Lam, R. Huang, and P.-S. Cheung. Learning phonetic similarity for matching named entity translations and mining new translations. In *Proceedings of the 27th Annual International ACM SIGIR Conference*, 2004.

[18] W. H. Lin and H. H. Chen. Backward machine transliteration by learning phonetic similarity. In *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL)*, pages 139–145, 2002.

[19] W.-H. Lu, L.-F. Chien, and H.-J. Lee. Mining anchor texts for translation of Web queries. *ACM Transactions on Asian Language Information Processing*, 1(2):159–172, 2002.

[20] J. Nie, M. Simard, P. Isabelle, and R. Durand. Cross-language information retrieval based on parallel texts and automatic mining of parallel

texts from the Web. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 74–81, 1999.

[21] R. Rapp. Automatic identification of word translations from unrelated English and German corpora. In *Proceedings of ACL-99*, pages 519–526, 1999.

[22] P. Thompson and C. Dozier. Name searching and information retrieval. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 134–140, 1997.

[23] E. Voorhees and D. Tice. The TREC-8 question answering track evaluation. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, 2000.

[24] B. Widrow and M.E. Hoff. Adaptive switching circuits. *1960 IRE WESCON Convention Record*, pages 96–104, 1960.

[25] S. Ye, T.S. Chua, and J. Liu. An agent-based approach to Chinese named entity recognition. In *Proceedings of the International Conference on Computational Linguistics*, pages 1149–1155, 2002.

[26] B. Zhao and S. Vogel. "Adaptive Parallel Sentences Mining from Web Bilingual News Collection". In *Proceedings of IEEE International Conference on Data Mining (ICDM)*, pages 745–748, December 2002.

Appendix A

This appendix depicts a table showing the pronunciation symbols used by the PRONLEX lexicon.

PRONLEX pronunciation symbol	Examples
i	heed, heat, he
I	hid, hit
e	aid, hate, hay
E	head, bet
@	had, hat
a	hod, hot
c	law, awe
o	hoed, oats, owe
U	could, hood
u	who'd, hoot, who
Y	hide, height, high
O	Boyd, boy
W	how'd, out, how
R	father(2); herd, hurt, her
x	data (2);
A	cud, bud
H	which
w	witch
y	yes
r	Ralph
l	lawn

Table A.1: The pronunciation symbols used by PRONLEX and some examples

PRONLEX pronunciation symbol	Examples
m	me
n	no
N	button(2)
G	hang
p	pot
b	bed
t	tone
d	done
k	kid
g	gaff
C	check
J	judge
f	fix
v	vex
T	thin
D	this
s	six
z	zoo
S	shin
Z	pleasure(2)
h	help

Table A.2: (continue) The pronunciation symbols used by PRONLEX and some examples

Appendix B

This appendix depicts a table showing the most likely tag of each English letter used in the transformation-based error-driven learning process in Chapter 4.1.

English letter	Most likely tag
-	end
a	@
b	b
c	k
d	d
e	end
'	end
f	f
g	end
h	end
i	I
j	J
-	end
k	k
l	l
.	end
m	m
n	n
o	o
p	p
q	k
r	r
s	s
t	t
u	A
v	v
w	w
x	ks
y	i
z	z

Table B.1: The most likely tag of each English letter

Appendix C

This appendix depicts a table showing the unseen name translations discovered from Day 2 to Day 28 , which is the experimental result discussed in Chapter 8.

Day	Discovered new unseen name translations
2	謝瓦爾德納澤/Shevardnadze, 鮑切/Boucher, 聯合國/UN, 拉希姆/Rahim Franke, 弗蘭克/Rahim Franke
3	鮑切爾/Powell, 拉夫羅夫/Sergei Lavrov, 捷夫扎澤/Tevzadze, 塔拉巴尼/Tony Blair
4	薩哈希維利/Mikhail Saakashvili, 迪尼/Daniel, 薩阿卡斯維利/Mikhail Saakashvili, 薩阿敦/Saddam, 布爾扎納澤/Nino Burjanadze, 佩羅尼/Blair, 謝瓦納茲/Shevardnadze
5	薩卡什維利/Mikhail Saakashvili, 阿拉伯聯合酋長國/Arab, 布朗/Bryan
6	巴格達市/Baghdad, 布隆克特/David Blunkett, 布雅納茲/Nino Burjanadze, 西斯塔尼/Sistani, 塔拉巴尼/Talabani, 奇切克/Cicek, 台北市/Taipei, 車臣/Chechnya, 拉姆斯菲爾德/Donald H. Rumsfeld
7	布萊爾/Powell, 布魯/Bill, 阿卜杜勒拉扎克/Abderrazak, 阿卡杜勒扎伊克/Abderrazak, 弗拉季米爾/Vladimir Putin, 阿里/Al
8	戴安娜/Diana, 耶爾馬茲/Yilmaz, 海珊/Hussein, 艾布尼/Allen Abney, 羅哈尼/Mr Rohani
9	穆沙拉夫/Pervez Musharraf, 穆夏拉夫/Pervez Musharraf, 索非亞/Syria, 也門/Amman
10	巴基斯坦/Pakistani, 佩斯利/Ian Paisley, 塞夫/Chavez, 伊安/Ian, 阿洛尼/Allen Abney, 莫頓/Madrid, 提姆/Tommy Thompson
11	菲爾/Phil, 博爾頓/John Bolton
12	穆拉/Mori, 桑迪/Saddam, 阿茲納爾/Aznar, 瓦杰帕伊/Behari Vajpayee
13	佐利克/Robert Zoellick, 扎拉特/Izzat Ibrahim
14	雷德/Reid
15	鮑威爾/Colin Powell, 拉布/Jiaboa
16	穆加貝/Mugabe, 河北省/Hebei, 新華社/Xinhua, 阿雷曼/Arnoldo Aleman, 侯賽因/Hussein
17	弗拉吉米爾/Vladimir, 尼亞利夫/Luzhkov, 亞博盧/Yabloko, 克里姆林宮/Pro Kremlin United
18	安南/Kofi Annan, 阿拉/Ariel, 阿呂/Chan Lien, 普京/Putin
19	布里茨/Blunt Bush, 斯特/Shiite, 桑切斯/Ricardo Sanchez, 家寶/Jiabao
20	克雷革安/Jean Chretien
21	溫家寶/Wen Jiabao
22	布雷默/Paul Bremer, 地中海/Mediterranean, 布拉希米/Brahimi, 布魯斯/Bush
23	薩達姆/Sadam, 瑪麗塔/Myard
24	墨西哥城/Mexico, 克雷蒂/Kellogg, 奧迪爾諾/Ray Odierno, 提克里蒂/Tikrit, 亞娜/Eli, 希拉克/Chirac
25	莎尼/Sunni, 羅伯茨/Robertson, 雷蒙德/Raymond
26	哈桑/Hussein, 遜尼派/Sunni, 亞娜/Annan
27	貝盧斯科尼/Silvio Berlusconi, 貝魯特/Bennett, 卡倫/Charles Cullen, 查爾斯/Charles Cullen
28	陳金/Changi

Table C.1: The unseen name translations discovered from Day 2 to Day 28

Appendix D

This appendix shows the experiment results for the gloss translation performance of the context-based model, equal-weighting model, and usage-factor model on the Mandarin news in the TDT-3 corpus.

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.4989	0.3481	0.4043
0.7	0.4853	0.3603	0.4073
0.5	0.4703	0.3750	0.4107
0.4	0.4610	0.3838	0.4120
0.3	0.4493	0.3942	0.4130
0.1	0.4153	0.4226	0.4114
0.03	0.3956	0.4379	0.4079
0.001	0.3815	0.4478	0.4041

Table D.1: The gloss translation performance of the context-based model on the Mandarin news in the TDT-3 corpus with window size of 1

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.4911	0.3801	0.4236
0.7	0.4766	0.3944	0.4263
0.5	0.4601	0.4116	0.4289
0.4	0.4502	0.4220	0.4299
0.3	0.4378	0.4339	0.4298
0.1	0.4022	0.4642	0.4247
0.03	0.3823	0.4793	0.4190
0.001	0.3685	0.4890	0.4140

Table D.2: The gloss translation performance of the context-based model on the Mandarin news in the TDT-3 corpus with window size of 2

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.4922	0.3858	0.4279
0.7	0.4767	0.4008	0.4304
0.5	0.4596	0.4186	0.4328
0.4	0.4493	0.4294	0.4336
0.3	0.4367	0.4419	0.4336
0.1	0.4008	0.4727	0.4278
0.03	0.3814	0.4879	0.4220
0.001	0.3673	0.4975	0.4165

Table D.3: The gloss translation performance of the context-based model on the Mandarin news in the TDT-3 corpus with window size of 3

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.6538	0.1350	0.2201
0.7	0.6538	0.1350	0.2201
0.5	0.6538	0.1350	0.2201
0.4	0.5378	0.3343	0.4073
0.3	0.4479	0.4373	0.4374
0.1	0.3436	0.5504	0.4189
0.03	0.3404	0.5519	0.4169
0.001	0.3404	0.5519	0.4169

Table D.4: The gloss translation performance of the equal-weighting model on the Mandarin news in the TDT-3 corpus

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.5806	0.0674	0.1190
0.7	0.5794	0.1062	0.1767
0.5	0.5438	0.1325	0.2100
0.4	0.4747	0.2365	0.3115
0.3	0.4162	0.3165	0.3554
0.1	0.2981	0.4657	0.3598
0.03	0.2145	0.5405	0.3039
0.01	0.1190	0.6298	0.1985

Table D.5: The gloss translation performance of the usage-factor model on the Mandarin news in the TDT-3 corpus

Appendix E

This appendix shows the experiment results for the gloss translation performance of the context-based model, equal-weighting model, and usage-factor model on the Mandarin news in the TDT-4 corpus.

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.5207	0.3363	0.4023
0.7	0.5065	0.3472	0.4051
0.5	0.4907	0.3608	0.4084
0.4	0.4813	0.3688	0.4100
0.3	0.4693	0.3784	0.4110
0.1	0.4342	0.4054	0.4105
0.03	0.4134	0.4201	0.4077
0.001	0.3989	0.4296	0.4045

Table E.1: The gloss translation performance of the context-based model on the Mandarin news in the TDT-4 corpus with window size of 1

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.5183	0.3546	0.4154
0.7	0.5044	0.3667	0.4186
0.5	0.4876	0.3813	0.4213
0.4	0.4776	0.3902	0.4225
0.3	0.4655	0.4006	0.4233
0.1	0.4304	0.4276	0.4210
0.03	0.4110	0.4411	0.4173
0.001	0.3986	0.4493	0.4142

Table E.2: The gloss translation performance of the context-based model on the Mandarin news in the TDT-4 corpus with window size of 2

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.5134	0.3780	0.4305
0.7	0.4974	0.3917	0.4330
0.5	0.4796	0.4081	0.4351
0.4	0.4688	0.4178	0.4357
0.3	0.4558	0.4292	0.4357
0.1	0.4182	0.4592	0.4308
0.03	0.3978	0.4743	0.4257
0.001	0.3832	0.4835	0.4205

Table E.3: The gloss translation performance of the context-based model on the Mandarin news in the TDT-4 corpus with window size of 3

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.4698	0.1812	0.2515
0.7	0.4698	0.1812	0.2515
0.5	0.4698	0.1812	0.2515
0.4	0.47532	0.3614	0.4046
0.3	0.4228	0.4500	0.4311
0.1	0.3417	0.5517	0.4183
0.03	0.3382	0.5536	0.4162
0.001	0.3382	0.5536	0.4162

Table E.4: The gloss translation performance of the equal-weighting model on the Mandarin news in the TDT-4 corpus

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.6117	0.0632	0.1131
0.7	0.5897	0.0985	0.1665
0.5	0.5563	0.1268	0.2043
0.4	0.4937	0.2291	0.3094
0.3	0.4317	0.3067	0.3546
0.1	0.3078	0.4569	0.3639
0.03	0.2217	0.5353	0.3102
0.01	0.1260	0.6245	0.2078

Table E.5: The gloss translation performance of the usage-factor model on the Mandarin news in the TDT-4 corpus

Appendix F

This appendix shows the experiment results for the gloss translation performance of the context-based model and equal-weighting model on the Arabic news in the TDT-3 corpus.

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.2220	0.4351	0.2930
0.7	0.2058	0.4724	0.2858
0.5	0.1872	0.5145	0.2735
0.4	0.1761	0.5376	0.2643
0.3	0.1632	0.5627	0.2521
0.1	0.1275	0.6214	0.2110
0.03	0.1085	0.6458	0.1854
0.001	0.0929	0.6601	0.1626

Table F.1: The gloss translation performance of the context-based model on the Arabic news in the TDT-3 corpus with window size of 3

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.4613	0.0737	0.1243
0.7	0.4613	0.0737	0.1244
0.5	0.4607	0.0741	0.1244
0.4	0.3484	0.1701	0.2246
0.3	0.2713	0.2324	0.2461
0.1	0.0967	0.5826	0.1654
0.03	0.0832	0.6698	0.1477
0.001	0.0832	0.6698	0.1477

Table F.2: The gloss translation performance of the equal-weighting model on the Arabic news in the TDT-3 corpus

Appendix G

This appendix shows the experiment results for the gloss translation performance of the context-based model and equal-weighting model on the Arabic news in the TDT-4 corpus.

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.3490	0.4911	0.4062
0.7	0.3254	0.5256	0.4000
0.5	0.3001	0.5599	0.3886
0.4	0.2820	0.5790	0.3771
0.3	0.2630	0.5999	0.3636
0.1	0.2308	0.6489	0.3389
0.03	0.2283	0.6505	0.3365
0.001	0.2283	0.6505	0.3365

Table G.1: The gloss translation performance of the context-based model on the Arabic news in the TDT-4 corpus with window size of 3

Translation weight cutting threshold	Precision	Recall	F-Measure
0.9	0.3959	0.0585	0.0998
0.7	0.3956	0.0586	0.0999
0.5	0.3954	0.0591	0.1001
0.4	0.3040	0.1449	0.1923
0.3	0.2359	0.2091	0.2175
0.1	0.0899	0.5730	0.1549
0.03	0.0765	0.6677	0.1370
0.001	0.0765	0.6677	0.1370

Table G.2: The gloss translation performance of the equal-weighting model on the Arabic news in the TDT-4 corpus