Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 183-200

An Approach to Using the Web as a Live Corpus for Spoken Transliteration Name Access

Ming-Shun Lin*, Chia-Ping Chen+ and Hsin-Hsi Chen*

Abstract

Recognizing transliteration names is challenging due to their flexible formulation and lexical coverage. In our approach, we employ the Web as a giant corpus. The patterns extracted from the Web are used as a live dictionary to correct speech recognition errors. The plausible character strings recognized by an Automated Speech Recognition (ASR) system are regarded as query terms and submitted to Google. The top *N* snippets are entered into PAT trees. The terms of the highest scores are selected. Our experiments show that the ASR model with a recovery mechanism can achieve 21.54% performance improvement compared with the ASR only model on the character level. The recall rate is improved from 0.20 to 0.42, and the MRR from 0.07 to 0.31. For collecting transliteration names, we propose a named entity (NE) ontology generation engine, called the X_{NE}-*Tree* engine, which produces relational named entities by a given seed. The engine incrementally extracts high co-occurring named entities with the seed. A total of 7,642 named entities in the ontology were initiated by 100 seeds. When the bi-character language model is combined with the NE ontology, the ASR recall rate and MRR are improved to 0.48 and 0.38, respectively.

1. Introduction

Named entities [MUC 1998], which denote persons, locations, organizations, etc., are common foci of searchers. Thompson and Dozier [1997] showed that named entity recognition (NER) could improve the performance of information retrieval systems. Capturing named entities is challenging due to their flexible formulation and novelty. The issues behind speech recognition make named entity recognition more challenging on the

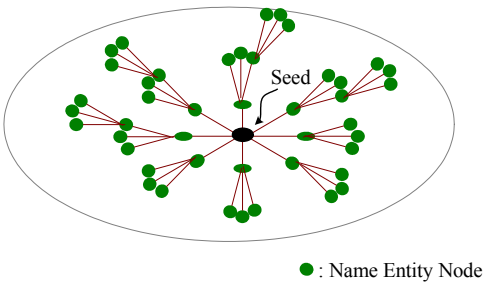
* Department of Computer Science and Information Engineering, National Taiwan University, Taipei, 106 Taiwan
E-mail: {d91022 , hhchen}@csie.ntu.edu.tw
+ Department of Computer Science and Engineering, National Sun Yat-Sen University, 70, Lien-Hai Road, Kaohsiung, 804 Taiwan
E-mail: cpchen@cse.nsysu.edu.tw

[Received February 16, 2006; Revised June 23, 2006; Accepted July 10, 2006]

spoken level than on the written level. This paper focuses on a special type of named entities, called transliteration names. They describe foreign people, places, etc. Spoken transliteration name recognition is useful for many applications. For example, cross language image retrieval via spoken queries aims to employ the latter in one language to retrieve images with captions in another language [Lin *et al*. 2004].

In the past, Appelt and Martin [1999] adapted the TextPro system to process transcripts generated by a speech recognizer. Miller *et al.* [2000] analyzed the effects of out-of-vocabulary errors and the loss of punctuation on name finding in automatic speech recognition. Huang and Waibel [2002] proposed an adaptive method of named entity extraction for the meeting understanding. Chen [2003] dealt with spoken cross-language access to image collections. The coverage of a lexicon is one of the major issues in spoken transliteration name access. Recently, researchers are interested in using the Web, which provides a huge collection of up-to-date data, as a corpus. Keller and Lapata [2003] employed the Web to obtain frequencies for bigrams that are unseen in a given corpus.

Named entities are important objects in web documents. Building named entity relationship chains from the web is an important task. Matsuo *et al*. [2004] found social networks of trust from related web pages. Google sets[1] extracts named entity from web pages by inputting a few named entities. For some emerging applications like personal name disambiguation [Fleischman and Hovy 2004] [Mann and Yarowsky 2003], social chain finding [Bekkerman and McCallum 2005] [Culotta *et al*. 2004] [Raghavan *et al*. 2004], etc., glossary-based representations of named entities are not enough. For collecting transliteration names and building a bi-character language model, we propose a named entity (NE) ontology generation engine, called the X_{NE}-*Tree* engine. This engine produces relational named entities by given a seed. The engine uses Google to incrementally extract high co-occurrence named entities from related web pages and those named entities have similar relational properties with the seed. In each iterative step, the seed will be replaced by its siblings or descendants, which form new seeds. In this way, the X_{NE}-*Tree* engine will build a tree structure as follows with the original seed as a root.



[1] http://labs.google.com/sets

In this paper, we discuss using the Web as a live dictionary for recognizing spoken transliteration names and employ the fuzzy search capability of Google to retrieve relevant web page summaries. In section 2, we sketch the steps in our method. In section 3, we discuss using PAT trees to learn patterns from the Web dynamically and to correct recognition errors. Section 4 shows the experiments, which are the ASR model with/without the recovery mechanism. Section 5 presents the X_{NE}-*Tree* named entity ontology engine and our experimental results. In section 6, we make concluding remarks.

2. Spoken Transliteration Name Recognition System

The spoken transliteration name recognition system shown in Figure 1 accepts a speech signal denoting a foreign named entity and converts it into a character string. It is composed of the following four major stages. Stages (1) and (2) consist of the fundamental tasks in speech recognition. In the Stages (3) and (4), speech-to-text errors are corrected by using the Web.

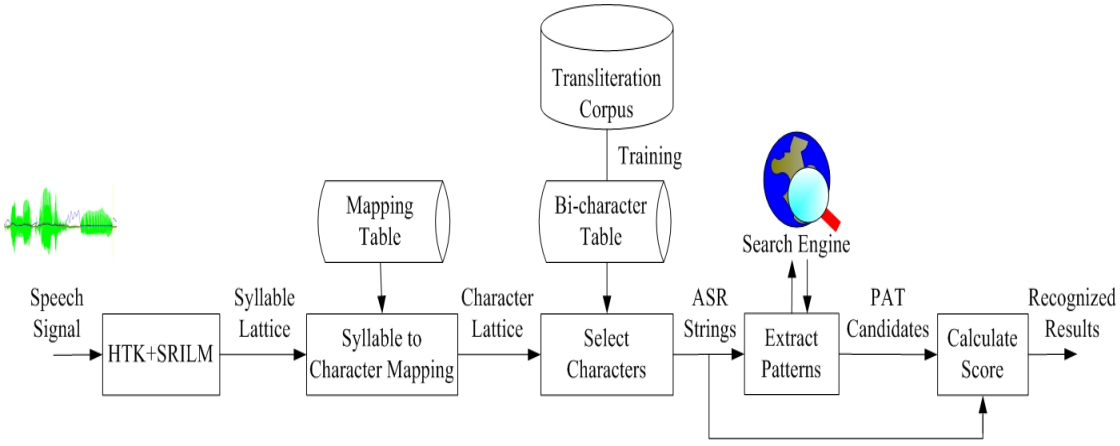


Figure 1. Stage in transliteration name recognition

(1) First, we employ the HTK[2] and SRILM[3] toolkits to build speech recognition models. For each speech signal, we use the model to get a syllable lattice.

(2) Then, the syllable lattice is mapped into a character lattice by using a mapping table. The mapping table is a syllable-to-character mapping. Top-*N* character strings are selected from the character lattice by using Viterbe algorithm and a bi-character model which is trained from a transliteration name corpus. Such character strings will be called *ASR strings* in the following.

(3) Next, each ASR string is regarded as a query and is submitted to a web search engine like Google. From the top-*N* search result, we select higher frequency patterns from a PAT tree structure. The PAT tree [Chien 1997] [Gonnet *et al*. 1992], which was derived from the Patricia

[2] http://htk.eng.cam.ac.uk/

[3] http://www.speech.sri.com/projects/srilm/

tree, can be employed to extract word boundary and key phrases automatically. Because we employ the PAT tree to extract patterns, the patterns will be called *PAT candidates* in the following. A PAT tree example, "湯姆漢克斯湯姆克魯斯喬治克魯尼" in MS950 encoding, is shown in Figure 2. The circles represent semi-infinite string numbers. The number above each circle denotes the length, which indicates the first different bit of the character strings recorded in the sub-trees. In this example, the longest patterns are for "克魯" and "湯姆" on nodes (7, 12) and (0, 5), with lengths of 33 and 34 bits, respectively. The second longest patterns are for "克", "魯", "姆" and "斯" on nodes (3, 7, 12), (8, 13), (1, 6) and (4, 9), with lengths of 16, 17, 18 and 18 bits, respectively.

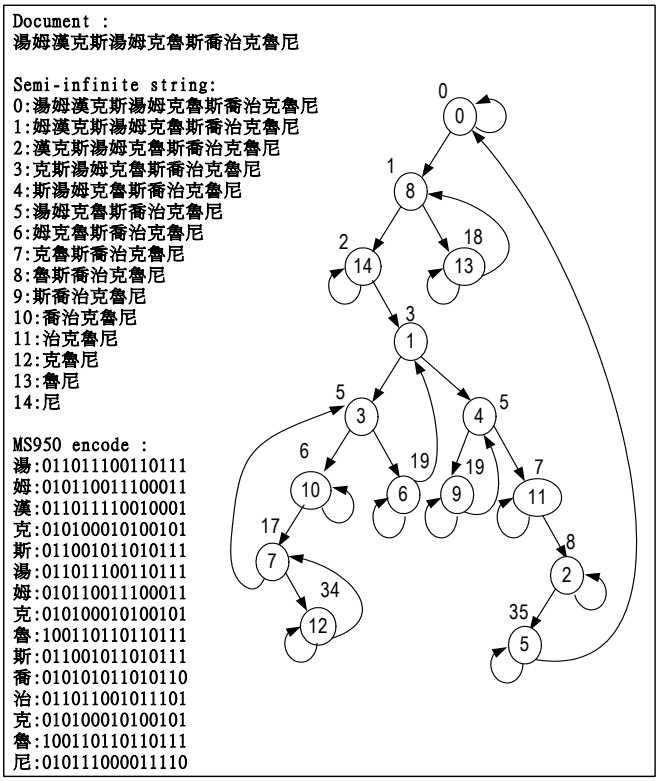


Figure 2. An example of extracting longest length pattern and its frequency

(4) Finally, the PAT candidates of all the ASR strings are merged together and ranked based on their number of occurrences and similarity scores. Candidates with the highest ranks are regarded as the recognition results for a spoken transliteration name.

Consider the example shown in Figure 3. The Chinese speech signal is a transliteration name, "湯姆克魯斯", in Chinese, which denotes the name of the movie star "Tom Cruise." The lattice shows different combinations of syllables. Each syllable corresponds to several Chinese characters. For example, "ke" is converted into "克", "柯", "科", "可", "喀", "刻", etc. The ASR strings "塔莫克魯斯", "塔門克魯斯", "塔莫柯魯斯", etc. are selected from the

character lattice. Through Google fuzzy search using the query "塔莫克魯斯", some summaries of Chinese web pages are obtained and shown in Figure 4. Although the common transliteration of "Tom Cruise" in Chinese is "湯姆克魯斯", which is different from the query "塔莫克魯斯", fuzzy matching using Google can still identify relevant snippets containing the correct transliteration. We will call this operation "recognition error recovery using the Web" in the following.

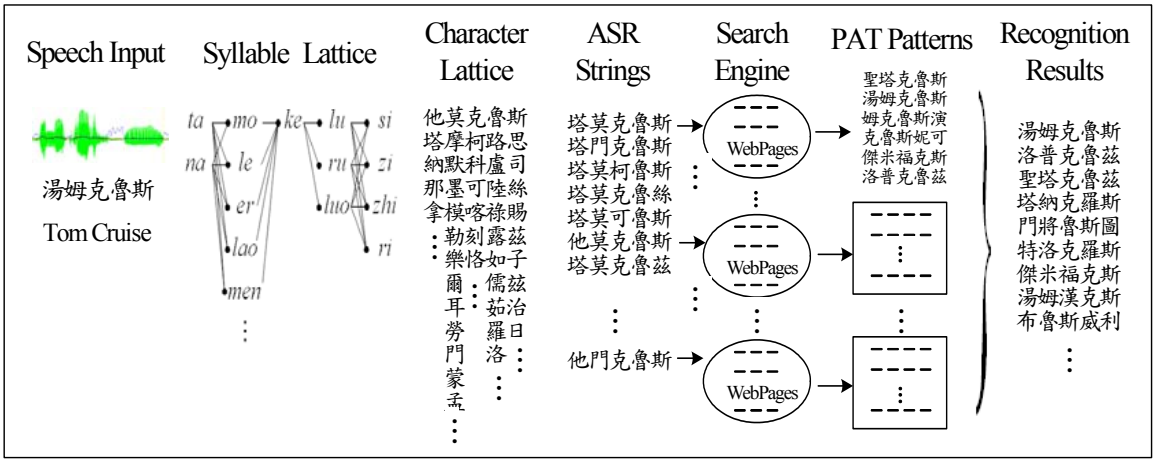


Figure 3. An example of recognizing a transliteration name: "湯姆克魯斯" ("Tom Cruise")

(1) ... 至於贏家部份，則還是湯姆漢克斯、湯姆克魯斯、喬治克魯尼這些老面孔，...

(2) ... 第 76 分鐘，克魯斯換下梅開二度的維埃裏。

(3) ... 國際米蘭 (4-4-2)：豐塔納/科爾多瓦，布爾迪索，馬特拉齊，法瓦利/斯坦科維奇，貝隆，扎內蒂，埃姆雷/克魯斯，馬丁斯。

(4) ... 提起妮可即發火湯姆克魯斯"想殺死記者".

(5) ... 電影節最具看點的明星當然非妮可基德曼與湯姆克魯斯有望在水城的戲劇性重逢莫屬。

Figure 4. Summaries obtained through fuzzy search for the query "塔莫克魯斯"

In the above examples, each partial matching part is enclosed in a rectangle symbol and the correct transliteration name is underlined. Summaries (1), (4) and (5) mention the movie star "湯姆克魯斯" (Tom Cruise) and summaries (2) and (3) mention a football star, "克魯斯" (Cruz). Figure 3 shows that PAT patterns like "聖塔克魯斯", "湯姆克魯斯", "姆克魯斯演", etc. are proposed. After merging and ranking are performed, the possible recognition results are "湯姆克魯斯", "洛普克魯茲", "聖塔克魯茲", etc.

3. Recognition Error Recovery Using the Web

The error recovery module tries to select higher frequency patterns from the Web search results and substitute the speech recognition results of Stages 1 and 2 (shown in Section 2) with the pattern. In this approach, Web search results obtained with an ASR string are placed in a PAT tree, and PAT candidates are selected from the tree. Two points are worth noting. A PAT candidate should occur many times in the PAT tree and should be similar to the ASR string.

The frequency, *Freq*, of a *PAT* candidate can be computed easily based on the PAT tree structure. The similarity between a *PAT* candidate and an *ASR* string is modeled by edit distance, which is the minimum number of insertions, deletions and substitutions needed to transform one character string (*ASR*) into another string (*PAT*). The smaller number is, the more similar they are. The similarity score, between an *ASR* string and a *PAT* string, is the frequency of the *PAT* string minus their number of edit operations. Finally, the score of a *PAT* string relative to an *ASR* string is defined as follows:

$$Score(ASR, PAT) = \alpha \times Freq(PAT) - \beta \times Dis\tan ce(ASR, PAT). \quad (1)$$

It is computed through weighted merging of the frequency of the *PAT* string and by using the similarity between the *ASR* string and *PAT* string. This value determines if the *ASR* string will be replaced by the *PAT* string. In the above example, *Freq*(湯姆克魯斯)=43 and *Distance* (塔莫克魯斯, 湯姆克魯斯)=2.

4. Experimental Results

The speech input to the transliteration name recognition system is a Chinese utterance. We employed 51,111 transliteration names [Chen *et al*. 2003] to train the bi-character language model discussed in Section 2. In the experiments, the test data include 50 American state names, 29 names of movie stars from the 31st Annual People's Choice Awards (http://www.pcavote.com), and 21 names of NBA stars from the 2005 NBA All Star Team (http://www.nba.com/allstar2005/). The 50 American state names are not very active on the Web. In contrast, the 50 names of stars are very active. The test set is different from the training data set, so it is an open test. Because there may be more than one transliteration for a foreign named entity, the answer keys are manually prepared. For example, "Arizona" has four possible transliterations in Chinese: "亞利桑納**",** **"**亞歷桑納"**,** "亞利桑那", and "亞歷桑那". On average, there are 1.9 Chinese transliterations for a foreign name in our test set. Appendix A lists the name test set and its answer keys. As explained in Section 2, the transliteration name recognition system is composed of four major stages. Stages 1 and 2 include the fundamental speech recognition tasks, and Stages 3 and 4 comprise the error

recovery task. To examine the effects of these two parts, we will evaluate them separately in the following two subsections.

4.1 Performance in the Error Recovery Task

We assume that correct syllables have been identified in the speech recognition task. We simulate this assumption by transforming all the characters in the answer keys into syllables. Then, in Stage 2, we map the syllable lattice to obtain a character lattice. A total of 50 ASR strings are extracted from the character lattice in Stage 2 and submitted to Google. Finally, the best 10 PAT candidates are selected. We use the MRR (Mean Reciprocal Rank) [Voorhees 1999] and recall rate to evaluate the performance. The MRR represents the average rank of the correctly identified transliteration names among in the proposed candidates and it is defined as follow:

$$MRR = \frac{1}{M}\sum_{i=1}^{M} r_i \text{ ,} \tag{2}$$

where M is the total number of test cases ; r_i equals $1/rank_i$ if $rank_i > 0$ and r_i is 0 if no answer is found. The $rank_i$ is the rank of the first correct answer for the i^{th} test case. That is, if the first correct answer is ranked 1, then the score is 1/1; if it is ranked 2, the score is 1/2, and so on. The MRR value is between 0 and 1. The inverse of the MRR denotes the average position of the correct answer in the proposed candidate list. The higher the MRR value is, the better the performance is. The recall rate is the number of correct references divided by M. It indicates how many transliteration names are correctly recognized.

Table 1. Performance of models wo/with error recovery

Models	Recall	MRR
ASR only	0.79	0.50
ASR + Web	0.90	0.88
ASR/Pre-removed + Web	0.59	0.48

Table 1 summarizes the experimental results obtained with models without/with the error recovery procedure. With the "ASR only" model, the top 10 ASR strings produced in Stage 2 are regarded as answers. This model does not employ the error recovery procedure. The recall rate is 0.79 and the MRR is 0.50. That is, 79 of 100 transliteration names are recognized correctly, and they appear in the first 2 (=1/0.50) position. In contrast, the "ASR + Web" model utilizes the error recovery procedure. PAT candidates extracted from the Web are selected in Stage 4. The recall rate is 0.90 and the MRR is 0.88. A total of 90 transliteration names are recognized correctly, and they appear in the first 1.13 (=1/0.88) position on average. In other words, when they are recognized correctly, they are always the top 1. Compared with

the first model, the recall rate is increased 13.92%. As for the third model, i.e., the "ASR/Pre-removed + Web" model, we try to evaluate the error recovery ability. The correct transliteration names appearing in the set of ASR strings are removed. That is, all of the ASR strings contain at least one incorrect character. In such cases, the recall rate is 0.59 and the MRR is 0.48. This means that 59 transliteration names are recovered, and they appeared in the first 2.08 (=1/0.48) position on average. We further examine the number of errors produced by the "ASR/Pre-removed + Web" model to study the error tolerance when using the Web. Table 2 shows the lengths of the transliteration names (in the rows), and the number of matching characters (in the columns). For a transliteration name of length *l*, the number of matching characters is 0 to *l*. Each cell denotes how many strings belong to the specific category. For example, before error recovery, there are 6, 25, 90, 184, and 0 strings of length 4, which have 0, 1, 2, 3, and 4 characters matching the corresponding answer keys, respectively. After error recovery, there are 19, 52, 66, 62, and 106 strings of length 4, which have 0, 1, 2, 3, and 4 characters matching the answer keys, respectively. In other words, the recovery procedure corrects some wrong characters. The number of 1-character (2-character) errors decreased from 184 (90) to 62 (66), and total number of correct strings are increased from 0 to 106.

Table 2. Distribution before/after error recovery

	Before Error Recovery							After Error Recovery						
Length of NEs	Number of Matching Characters							Number of Matching Characters						
	0	1	2	3	4	5	6	0	1	2	3	4	5	6
2	11	23	0	-	-	-	-	13	21	0	-	-	-	-
3	6	29	76	0	-	-	-	6	39	64	2	-	-	-
4	6	25	90	184	0	-	-	19	52	66	62	106	-	-
5	9	10	12	77	193	0	-	11	23	36	41	53	137	-
6	0	0	1	8	20	39	0	0	3	19	12	7	5	22

Table 3. Effects of error positions and string lengths

Error Positions	Length=2	Length=3	Length=4	Length=5	Length=6	Total
Position 1	0	0	37	42	7	86
Position 2	0	2	35	42	4	83
Position 3	-	0	20	19	9	48
Position 4	-	-	17	24	3	44
Position 5	-	-	-	14	3	17
Position 6	-	-	-	-	1	1
Total	0	2	109	141	27	279

Table 3 shows the effects of the error position (in the rows) and the string length (in the columns). A total of 0, 2, 106, 137, and 22 utterances recover 1 character with length 2, 3, 4, 5, and 6, respectively. A total of 0, 0, 3, 4, and 5 utterances recover 2 characters with length 2, 3, 4, 5, and 6, respectively. No utterances can recover over 3 characters. The cell denotes how many strings can be recovered under the specific position and length. For example, a total of 37, 35, 20, and 17 errors for strings of length 4 appearing at positions 1, 2, 3, and 4, respectively, can be recovered by using the Web. In the experiments, 0% (=0/34), 1.80% (=2/111), 35.74% (=109/305), 46.84% (=141/301), and 39.71% (=27/68) of the strings of length 2, 3, 4, 5, and 6 can be recovered, respectively. The 34 is the number of the PAT candidates with length 2. Similarly, the 111, 305, 301, and 68 are the number of the PAT candidates with length 3, 4, 5, and 6. As for length, the longer strings facilitate better recovery than the shorter strings. Another results show that 30.82% (=86/279), 29.75% (=83/279), 17.20% (=48/279), 15.77% (=44/279), 6.09% (=17/279), and 0.36% (=1/279) of the strings with incorrect character appearing at positions 1, 2, 3, 4, 5 and 6 can be recovered, respectively. The 279 is the number of characters on which the 100 test data. Because the bi-character language model proceeds from the left side to the right side, the errors occurring at the beginning are easier to recover than those at the end.

4.2 Performance in the Speech Recognition Task

The set of 100 transliteration names discussed in Section 4.1 are spoken by 2 males and 1 female, so 300 transliteration names are recorded. We employ HTK and SRILM to get the best 100 syllable lattices (N-Best, N=100). The TCC-300 dataset for Mandarin is used to train the acoustic models. There are 417 HMM models, and each one has 39 feature vectors. The syllable accuracy is computed as follows: (*M-I-D-S*)/*M* * 100%, where *M* is the number of correct syllables; *I*, *D*, and *S* denote the number of insertion, deletion, and substitution errors, respectively. The syllable accuracy is 76.57%. To estimate the character recovery ability, we consider the correct character number. The accuracy of the ASR only and ASR+Web models on the character level are computed as follows, respectively:

$$\sum_{i=1}^{M} \max_{j=1 to K} \left(\frac{Word_{Length}(TestName_i) - Distance(AnsKey_{ij}, ASR_i)}{Word_{Length}(TestName_i)} \right) \quad (3)$$

and

$$\sum_{i=1}^{M} \max_{j=1 to K} \left(\frac{Word_{Length}(TestName_i) - Distance(AnsKey_{ij}, PAT_i)}{Word_{Length}(TestName_i)} \right), \quad (4)$$

where *M* is the total test number and *K* is the answer key number for test name *i*. A total of 50

ASR strings are extracted from the character lattice, and the best 50 PAT candidates are selected. Table 4 shows the character level results. The "ASR+Web" model achieves 21.54% better performance than the "ASR Only" model on average. Table 5 shows the word level results. The "ASR+Web" model using error recovery procedure improves the recall rate and the MRR of the "ASR Only" model from 0.20 and 0.07 to 0.42 and 0.31, respectively. In other words, the average ranks of the correct transliteration names move from the 14th position (=1/0.07) to the 3rd position (=1/0.31) after error recovery.

Table 4. Performance on the character level

ASR Only (Character Level Accuracy)					ASR + Web (Character Level Accuracy)				
Top 1	Top 2	Top 3	Top 4	Top 5	Top 1	Top 2	Top 3	Top 4	Top 5
38.01%	43.34%	47.30%	49.07%	50.93%	48.18%	54.01%	55.93%	58.03%	59.48%

Table 5. Performance on the word level

ASR Only (Word Level)		ASR + Web (Word Level)	
Recall	MRR	Recall	MRR
0.20	0.07	0.42	0.31

Web fuzzy search produces useful patterns for error recovery. Our fault tolerance experiments show that longer transliteration names have stronger tolerance than shorter transliteration names and that the incorrect characters appearing at the beginning of a transliteration name are relatively easier to correct than those appearing at the end. Thus, the improvement in the character level accuracy is helpful for the recovery mechanism, and vice versa.

5. Re-training the Bi-Character Language Model

For collecting transliteration names to build a bi-character language model, we propose using a named entity (NE) ontology generation mechanism, called the X_{NE}-*Tree* engine. Given a seed, the engine incrementally extracts relational named entities with the seed from related web pages and the output is a tree structure. Each node in the structure is a named entity (NE).

5.1 A Named Entity Ontology Generation Engine

Recognizing a named entity and calculating the relational property score with a seed are two crucial tasks. Firstly, we submit the given seed to a search engine and select the top N returned snippets. Then, we use the suffix tree to extract possible patterns automatically. The patterns, which are extracted based on the global statistic, may be impacted by the frequency variance of patterns with the same substrings [Yang and Li 2002]. Because our aim is to generate named entities, most of the max-duplicated strings can be filtered out by using a named entity

recognition (NER) system [Chen *et al*. 1998]. The NER system will re-segment a candidate pattern to obtain some substrings and give each substring a part of speech (POS) and a possible name tag. If any substring is tagged as a location, an organization, or a person by using an NER-POS server [Chen *et al.* 1998], the candidate pattern is considered to be a named entity. Because prepositions frequently occur before/after a named entity, the suffix tree approach may introduce an incorrect boundary. Thus, we filter out substrings that have a preposition tag.

Secondly, we calculate a relational property score, called the *Co-Occurrence Double-Check score (CODC*, for each extracted name entity (denoted Y_i) with a seed (denoted X). We postulate that X and Y_i have a strong relationship if we can find Y_i from X (a forward process) and find X from Y_i (a backward process). The forward and backward processes form a double check operation. $CODC(X, Y)$ is defined as follows:

$$CODC(X,Y) = e^{\log\left(\frac{f(Y@X)}{f(X)} \times \frac{f(X@Y)}{f(Y)}\right)^{\alpha}}, \tag{5}$$

where $f(X@Y_i)$ is the total number of occurrences of X in the top N snippets when query Y_i is submitted to the search engine. Similarly, $f(Y_i@X)$ is the total number of occurrences of Y_i in the top N snippets for query X; $f(X)$ is the total number of occurrences of X in the top N snippets for query X, and $f(Y)$ is the total number of occurrences of Y in the top N snippets of query Y. In each iterative step, Y_i is added to a queue when the $CODC(X,Y_i)$ value is larger than a threshold θ. Then, we get a new seed X from the queue. The *CODC* measure is best when α=0.15. The overall process is shown in Figure 5.

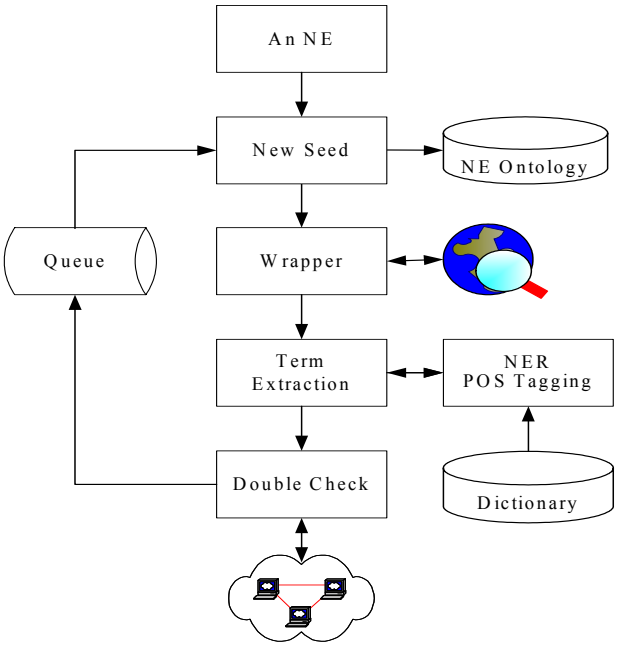


Figure 5. Named entity ontology generation process

5.2 Constructing a Named Entity Ontology

When building a bi-character language model, we choose 100 seeds, which are the same 100 utterances described in Section 4. Here, we set a condition to control generation of the ontology. Each initial seed can derive at most four layers, and no more than 15 children are allowed in the first layer. The maximum number of children of a named entity in layer i is bounded by the number in layer (i-1) multiplying a decreasing rate. In the experiments, we set the decreasing rate to be 0.7, so that at most 15, 15×0.7, 15×0.7^2, and 15×0.7^3 children can be expanded by a named entity in layers 0-3, respectively. We set the threshold θ at 0.1. Those named entities with *CODC* scores larger than the predefined threshold are sorted, and a sufficient number of named entities are selected in a sequence for expansion. In this way, a total of 7,642 nodes are generated by the 100 seeds. We employ Touch-Graph (http://www.touchgraph.com) to represent the named entity ontology. Figure 6 shows an example by using "湯姆克魯斯" as a seed, which is a Mandarin transliteration name of an actor "Tom Cruise", to build an ontology. To evaluate the performance, we consider the following four types.

(1) Named Entity (NE) type: In this case, the proposed candidate should be a named entity and should not have incorrect boundary. A personal name with a title or a first name with more than 4 characters is regarded as being correct. In contrast, patterns with a last name only are considered incorrect.

(2) Relational property of NE (RNE) type: For those acceptable strings in (1), which have the same relational property with the initial seed or its parents are considered to be correct. The remaining nodes are incorrect.

(3) Partial Named Entity (PNE) type: We relax the restriction on boundary errors specified in (1). Patterns consisting of partial named entities are regarded as being correct. The remaining nodes are incorrect.

(4) Relational property of PNE (RPNE) type: For those acceptable strings in (3), which have the same relational property with the initial seed or its parents are considered to be correct. The remaining nodes are incorrect.

Table 6 shows the performance in ontology generation. Of those 7,642 nodes, the error rates for the NE type, the RNE type, the PNE type, and the PRNE type are 19.60%, 34.20%, 12.62%, and 29.82%, respectively.

Table 6. Performance in ontology generation

Size of Seed	Size of Ontology	NE	RNE	PNE	RPNE
100	7,642	19.60%	34.20%	12.62%	29.82%

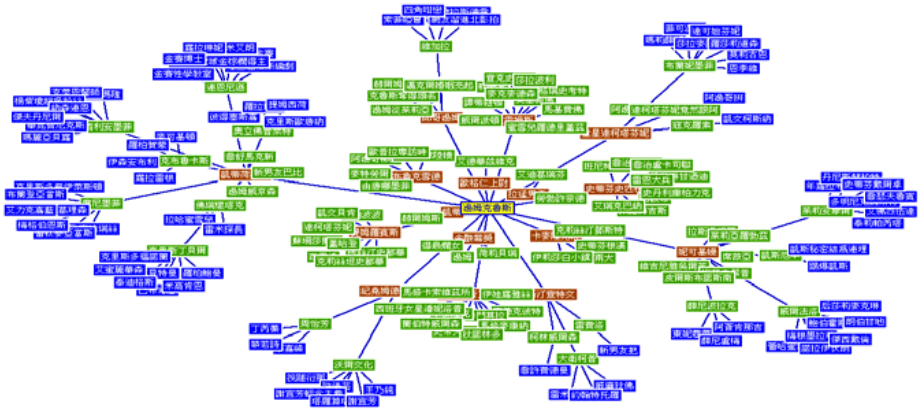


Figure 6. A snapshot of the named entity ontology of "湯姆克魯斯" ("Tom Cruise")

5.3 Combining the Bi-Character Language Model with the NE Ontology

In the previous experiments, we employed 51,111 transliteration names (BaselineTN) to build the bi-character language model. However, these transliteration names might not be active on the Web. We submitted these transliteration names to a search engine (i.e., Google). For a transliteration name, if the search engine does not return any web pages, we filter it out. Finally, we filter out 14,933 named entities and get 36,178 transliteration names (FilterTN) with this method. Refer to Table 6. Of the 7,642 named entities (Total-Ontology) reported by X_{ne}-engine, 6,146 named entities (NE-Ontology) are of the correct NE type, and 5,023 named entities (RNE-Ontology) are of the correct RNE type.

In the experiments, we consider word level accuracy only. Two basic transliteration name corpora, i.e., BaselineTN and FilterTN, are employed to build bi-character language models. In ideal case, correct syllables have been identified in the ASR (ASR_Perfect). Table 7 shows that FilterTN is a little better than BaselineTN. We further combine FilterTN with the NE ontology derived by the X_{NE}-*Tree* engine to perform evaluation. In this way, we employ the FilterTN+RNE-Ontology, FilterTN+NE-Ontology, and FilterTN+Total-Ontology to build bi-character language models. Table 7 summarizes the experimental results obtained with the language model with the NE ontology. The three models with the NE ontology outperform the baseline model. In particular, the NE ontology improve the recall rate and the MRR from 0.79 and 0.50 (BaselineTN) to 0.84 and 0.55 (FilterTN+RNE-Ontology), respectively. Table 8 lists the results obtained using both the NE ontology and error recovery procedure. The NE ontology is still helpful, in particular for the recall rate. In the best case, it improves the recall rate from 0.90 (BaselineTN) to 0.94 (FilterTN+RNE-Ontology). In summary, the model using NE ontology resources, the recall rate is improved 13.92%. On comparing the "FilterTN+RNE-Ontology" model with the ASR model without the error recovery procedure and NE ontology resources, the recall rate is improved 18.98%.

Table 7. Bi-character language models with the NE ontology but without error recovery.

Language Model	Size of TN	ASR_Perfect Only (Word Level)	
		Recall	MRR
BaselineTN	51,111	0.79	0.50
FilterTN	36,178	0.80	0.50
FilterTN + RNE-Ontology	41,201	0.84	0.55
FilterTN + NE-Ontology	42,324	0.83	0.57
FilterTN + Total-Ontology	43,820	0.82	0.57

Table 8. Bi-character language models with both the NE ontology and error recovery procedure

Language Model	ASR_Perfect + Web (Word Level)	
	Recall	MRR
BaselineTN	0.90	0.88
FilterTN	0.90	0.87
FilterTN + RNE-Ontology	0.94	0.88
FilterTN + NE-Ontology	0.93	0.88
FilterTN + Total-Ontology	0.93	0.90

Table 9. Combining the bi-character language model with the NE ontology without/with the error recovery procedure in ASR systems

Language Model	ASR Only (Word Level)		ASR+Web (Word Level)	
	Recall	MRR	Recall	MRR
BaselineTN	0.20	0.07	0.42	0.31
FilterTN	0.20	0.06	0.41	0.32
FilterTN + RNE-Ontology	0.23	0.11	0.48	0.38
FilterTN + NE-Ontology	0.24	0.11	0.48	0.37
FilterTN + Total-Ontology	0.24	0.12	0.47	0.39

Table 9 summarizes the experimental results obtained with language models that use the NE ontology without/with error recovery procedure in the complete transliteration name ASR system. The system without the error recovery procedure (ASR Only), the NE ontology still improves the performance. Comparing the "FilterTN+RNE-Ontology" with BaselineTN, the recall rate is increased 15%. When the ASR system incorporates the error recovery procedure (ASR+Web), the recall rate is increased 14.28% (FilterTN+RNE-Ontology vs. BaselineTN).

6. Conclusions

In this study, we employ the Web as a giant corpus to correct transliteration name recognition errors. Web fuzzy search produces useful patterns for error recovery. In the ideal case, we input the correct syllable sequences, convert them into text strings, and test the recovery capability by using the Web corpus. On comparing with the model without the web recovery procedure, the recall rate is improved 13.92%. For collecting transliteration names beforehand, we propose using a named entity (NE) ontology generation engine, called the X_{NE}-*Tree* engine. The engine automatically creates named entity ontology for a given seed. In the experiments, a total of 7,642 named entities in the ontology were initiated by 100 seeds. After the language model for speech recognition combined the named entity ontology, the recall rate is improved 18.98%. With a complete transliteration name ASR system, the error recovery experiments show that the recall rate is increased from 0.20 to 0.42 and the MRR from 0.07 to 0.31. When the RNE-Ontology is incorporated, the recall rate and the MRR is increased 0.48 and 0.38, respectively. Thus, we conclude that the error recovery procedure and NE ontology can helpful to the ASR model.

Acknowledgements

This research was partially supported by the National Science Council, Taiwan, under contracts NSC94-2752-E-001-001-PAE and NSC95-2752-E-001-001-PAE.

References

Appelt, D. E., and D. Martin, "Named Entity Extraction from Speech: Approach and Results Using the TextPro System," In *Proceedings of DARPA Broadcast News Workshop*, 1999, pp. 51-54.

Bekkerman, R., and A. McCallum, "Disambiguating Web Appearances of People in a Social Network," In *Proceedings of WWW*, 2005, pp. 463-470.

Chen, H. H., "Spoken Cross-Language Access to Image Collection via Captions," In *Proceedings of 8th Eurospeech*, 2003, pp. 2749-2752.

Chen, H. H., C. H. Yang, and Y. Lin, "Learning Formulation and Transformation Rules for Multilingual Named Entities," In *Proceedings of the Association for Computational Linguistics on Multilingual and Mixed-language Named Entity Recognition*, 2003, pp. 1-8.

Chen, H. H., Y. W. Ding, and S. C. Tsai, "Named Entity Extraction for Information Retrieval," *Computer Processing of Oriental Languages*, *Special Issue on Information Retrieval on Oriental Languages*, 1998, pp. 75-85

Chien, L. F., "PAT-Tree-Based Keyword Extraction for Chinese Information Retrieval," In *Proceedings of 20th ACM SIGIR Conference*, 1997, pp. 50-58.

Culotta, A., R. Bekkerman, and A. McCallum, "Extracting Social Networks and Contact Information from email and the Web," In *Proceedings of the First Conference on Email and Anti-Spam*, 2004.

Fleischman, M. B., and E. Hovy, "Multi-document Person Name Resolution," In *Proceedings of the Association for Computational Linguistics (ACL), Reference Resolution Workshop*, 2004. (presentation order: second)

Gonnet, G. H., R. A. Baeza-Yates, and T. Snider, "New Indices for Text: PAT Trees and PAT Arrays," *In Information Retrieval Data Structures Algorithms*, 1992, pp. 66-82.

Huang, F., and A. Waibel, "An Adaptive Approach of Name Entity Extraction for Meeting Application," In *Proceedings Of Human Language Technology Conference*, 2002.

Keller, F., and M. Lapata, "Using the Web to Obtain Frequencies for Unseen Bigrams," *Computational Linguistics*, 2003, pp. 459-484.

Lin, W. C., M. S. Lin, and H. H. Chen, "Cross-Language Image Retrieval via Spoken Query," In *Proceedings of RIAO*, 2004, pp. 524-536.

Mann, G. S., and D. Yarowsky, "Unsupervised Personal Name Disambiguation," In *Proceedings of Conference on Computational Natural Language Learning*, 2003.

Matsuo, Y., H. Tomobe, K. Hasida, and M. Ishizuka, "Finding Social Network for Trust Calculation," In *Proceedings of 16th European Conference on Artificial Intelligence*, 2004, pp. 510-514.

Miller, D., S. Boisen, R. Schwartz, R. Stone, and R. Weischedel, "Named Entity Extraction from Noisy Input: Speech and OCR," In *Proceedings of 6th Applied Natural Language Processing Conference*, 2000, pp. 316-324.

MUC Message Understanding Competition, http://www.itl.nist.gov/iaui/894.02/related_projects/muc/index.html, 1998.

Raghavan, H., J. Allan, and A. McCallum, "An Exploration of Entity Models, Collective Classification and Relation Description," In *Proceedings of LinkKDD*, 2004.

Thompson, P., and C. Dozier, "Name Searching and Information Retrieval," In *Proceedings of Second Conference on Empirical Methods in Natural Language Processing*, 1997, pp. 134--140.

Yang, W., and X. Li, "Chinese Keyword Extraction Based on Max-Duplicated Strings of the Documents," In *Proceedings of 25th ACM SIGIR Conference*, 2002, pp. 439-440.

Voorhees, E., "The TREC-8 Question Answering Track Evaluation," In *Proceedings of the 8th TREC*, 1999, pp. 23-37.

Appendix A

Transliteration name	Answer keys list	Transliteration name	Answer keys list
科羅拉多	克羅拉多 柯羅拉多 科羅拉多	喬治克隆尼	喬治克隆尼 喬治克龍尼 喬治柯隆尼
加利福尼亞	加里福尼亞 加利福尼亞 加利弗尼亞	丹佐華盛頓	丹佐華聖頓 丹佐華盛頓
喬治亞	喬治亞 喬治雅	湯姆克魯斯	湯姆克魯斯
密西根	密希根 密西根	強尼戴普	強尼戴普
阿拉斯加	阿拉斯加	湯姆漢克斯	湯姆漢克斯
北卡羅萊納	北卡羅萊納 北卡羅萊那 北卡羅來納 北卡羅來那	*芮妮齊薇格	芮妮齊薇格 芮尼齊維格
康乃狄克	康乃迪克 康乃狄克 康迺迪克	莎莉賽隆	莎莉賽隆 莎莉塞隆 沙莉賽隆
德拉瓦	德拉瓦	妮可基嫚	妮可基嫚 尼可基曼 妮可基曼
佛羅里達	佛羅里達 佛羅理達	茱莉安摩爾	朱利安摩爾 茱莉安摩爾 朱莉安摩爾
南卡羅萊納	南卡羅萊納 南卡羅萊那 南卡羅來納 南卡羅來那	茱莉亞羅勃茲	茱莉亞羅勃茲 朱利亞羅伯茲 朱利亞羅勃茲 朱莉亞羅伯茲 茱莉亞羅伯茲
*夏威夷	夏威夷 夏威宜	威爾史密斯	威爾史密斯
愛荷華	艾荷華 愛何華 愛荷華	維果莫天森	維果莫天森 維果摩天森 維果墨天森
愛達荷	艾達荷 愛達荷	麥特戴蒙	麥特戴蒙
伊利諾	伊利諾 伊立諾 依利諾	休傑克曼	休傑克曼 休杰克曼
*印地安那	印地安那 印地安納 印弟安納	托貝馬奎爾	托貝馬奎爾
堪薩斯	坎薩斯 堪薩斯	鄔瑪舒嫚	烏瑪舒曼 鄔瑪舒曼
肯塔基	肯塔基	琪拉奈特莉	琪拉奈特莉 奇拉奈特莉
路易斯安那	路易斯安納	凱特貝琴薩	凱特貝琴薩
麻薩諸塞	麻薩諸塞 馬薩諸塞 麻塞諸塞	荷莉貝瑞	荷莉貝瑞 荷利貝瑞
緬因	緬因	安潔莉娜裘莉	安潔莉娜裘莉
馬里蘭	馬里蘭 馬利蘭	柴克巴爾夫	柴克巴爾夫
亞利桑那	亞利桑納 亞歷桑納 亞利桑那 亞歷桑那	布萊德彼特	布萊德比特 布萊德彼特 布萊得比特 布萊得彼特
明尼蘇達	明尼蘇達 明尼蘇答	金凱瑞	金凱瑞
密蘇里	米蘇里 密蘇里	柯林法洛	柯林法洛 科林法洛
密西西比	密西西比	裘德洛	裘德羅 裘德洛
蒙大拿	蒙大納 蒙大那 蒙大拿	娜塔莉波曼	娜塔利波曼 娜塔莉波曼
內布拉斯加	內布拉斯加	凱特溫絲蕾	凱特溫斯雷 凱特溫斯蕾 凱特溫絲雷 凱特溫絲蕾
阿拉巴馬	阿拉巴馬	*珍妮佛嘉納	珍妮佛嘉納
北達科他	北達科他 北達科塔	*茱兒芭莉摩	茱兒芭莉摩
新罕布夏	新漢布夏 新罕布夏	姚明	姚明
紐澤西	紐澤西	俠客歐尼爾	俠克歐尼爾 俠客歐尼爾
*新墨西哥	新墨西哥	凱文賈奈特	凱文加奈特 凱文賈奈特
內華達	內華達	*崔西麥格瑞迪	崔西麥格瑞迪 崔西麥葛瑞迪
紐約	紐約	柯比布萊恩	柯比布萊恩 科比布萊恩
俄亥俄	俄亥俄	文斯卡特	文斯卡特 文思卡特
奧克拉荷馬	奧克拉荷馬 奧克拉荷瑪 奧克拉河馬	提姆鄧肯	提姆鄧肯
奧勒	奧勒岡	葛蘭特希爾	格蘭特希爾 葛蘭特希爾
賓汐法尼亞	賓希法尼亞 賓西法尼亞 賓夕凡	勒布朗詹姆斯	勒布朗詹姆斯 勒布郎詹姆斯

	尼亞		
*羅德島	羅德島	艾倫艾弗森	艾倫艾弗森 艾倫埃弗森 艾倫艾佛森
阿肯色	阿肯色 阿肯瑟 阿肯塞	*小歐尼爾	小歐尼爾
南達科他	南達科塔 南達科他 南達柯塔	拉希德華萊士	拉希德華萊士 拉西德華萊士
田納西	田納西 田那西	普林斯	普林斯
德克薩斯	德克薩斯 得克薩斯	賈米森	賈米森
*猶他	猶他	比盧普斯	比魯普斯 比盧普斯
佛蒙特	佛蒙特	斯托賈科維奇	斯托賈克維奇 斯托賈科維奇 斯托賈可維奇
維吉尼亞	維基尼亞 維吉尼亞 維吉尼雅	德克諾維茨基	德克諾維茨基 德克諾威茨基 德克諾維斯基
華盛頓	華聖頓 華盛頓 華勝頓	班華萊士	班華萊士 班華勒斯
西維吉尼亞	西維基尼亞 西維吉尼亞 西維吉尼雅	卡梅隆安東尼	卡梅隆安東尼 卡麥隆安東尼
威斯康辛	威斯康辛 威斯康新	斯塔德邁爾	斯塔德麥爾 斯塔德邁爾 斯塔達邁爾
懷俄明	懷俄明	基里連科	基里連科

* “印、島、兒、猶、小、芮” characters are not in training set and “尼、妮”, ” 辛、新” and “奇、琪” differentia of frequency is too high.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 201-222

An Empirical Study of Word Error Minimization Approaches for Mandarin Large Vocabulary Continuous Speech Recognition

Jen-Wei Kuo*+, Shih-Hung Liu*, Hsin-Min Wang+, and Berlin Chen*

Abstract

This paper presents an empirical study of word error minimization approaches for Mandarin large vocabulary continuous speech recognition (LVCSR). First, the minimum phone error (MPE) criterion, which is one of the most popular discriminative training criteria, is extensively investigated for both acoustic model training and adaptation in a Mandarin LVCSR system. Second, the word error minimization (WEM) criterion, used to rescore *N*-best word strings, is appropriately modified for a Mandarin LVCSR system. Finally, a series of speech recognition experiments is conducted on the MATBN Mandarin Chinese broadcast news corpus. The experiment results demonstrate that the MPE training approach reduces the character error rate (CER) by 12% for a system initially trained with the maximum likelihood (ML) approach. Meanwhile, for unsupervised acoustic model adaptation, MPE-based linear regression (MPELR) adaptation outperforms conventional maximum likelihood linear regression (MLLR) in terms of CER reduction. When the WEM decoding approach is used for *N*-best rescoring, a slight performance gain over the conventional maximum a posteriori (MAP) decoding method is also observed.

Keywords: Broadcast News, Continuous Speech Recognition, Discriminative Training, Minimum Phone Error, Word Error Minimization

* Graduate Institute of Computer and Information Engineering, National Taiwan Normal University, Taipei, Taiwan
E-mail: rogerkuo@iis.sinica.edu.tw
+ Institute of Information Science, Academia Sinica, Taipei, Taiwan

[Received February 23, 2006; Revised August 24, 2006; Accepted September 5, 2006]

1. Introduction

Due to advances in computer technology and the growth of the Internet, large volumes of multimedia content, such as broadcast news, lectures, voice mails, and digital archives continue to grow and fill our computers, networks, and lives. It is obvious that speech is the richest source of information for the large volumes of multimedia content; thus, associated speech processing technologies will play an increasingly important role in multimedia organization and retrieval in the future. Among these technologies, automatic speech recognition (ASR) has long been the focus of research in the speech processing community.

Automatic speech recognition is a pattern classification task that classifies sound segments into different linguistic categories based on the acoustic vector sequence extracted from the speech signal. Traditionally, in most pattern classification applications, the goal of classifier design is to reduce the probability of errors by using the minimum error rate (MER) criterion [Duda *et al.* 2000]. Under this paradigm, the problems of classifier optimization are resolved by minimizing the expected loss over the training data directly. The zero-one loss function, which simply assigns no loss to a correct classification and a unit loss to an error, is often employed for this purpose. For example, in ASR, a hypothesized word sequence containing one or more word errors, or a totally different sequence, as compared to the correct sequence, will incur the same amount of loss. However, the most common performance evaluation metrics adopted in ASR often consider individual word errors, instead of merely counting the string-level errors. The use of the zero-one loss function leads to a mismatch between classifier optimization and performance evaluation. In recent years, a common practice in ASR has been to replace the zero-one loss function with alternative loss functions that consider word- or phone-level errors. In practice, such improved loss functions can be used in both model parameter estimation (i.e., classifier optimization) and speech decoding.

In this paper, we present an empirical study of word error minimization approaches for Mandarin large vocabulary continuous speech recognition (LVCSR). The minimum phone error (MPE) criterion is extensively investigated in both acoustic model training and adaptation; while the word error minimization (WEM) criterion is exploited to rescore *N*-best word strings.

The remainder of the paper is organized as follows. In Section 2, the general background of the Bayes risk and overall risk criteria is given, and their use in ASR is explained. Section 3 presents the application of the MPE criterion for acoustic model training, and Section 4 describes its extension to unsupervised linear regression based acoustic model adaptation. The use of the WEM criterion for speech decoding is discussed in Section 5. The experiment setup is detailed in Section 6 and a series of speech recognition experiments is described in Section 7. Finally, we present the conclusions drawn from the research in Section 8.

2. Bayes Risk and Overall Risk

Given an acoustic vector sequence O, the goal of an ASR system is to make a decision $\alpha_u(O)$ that identifies O as a certain word sequence u from a hypothesized space $\mathbf{W}_h$ of all possible word sequences in the language. Let $L(u,c)$ be the loss incurred by the decision $\alpha_u(O)$, where the correct (i.e., reference) transcription is c. Actually, we have no prior knowledge of the correct transcription; in other words, any arbitrary word sequence s in $\mathbf{W}_h$ could be identical to c. Consequently, for each possible decision $\alpha_u(O)$, the expected loss (or risk) is calculated as [Duda *et al.* 2000]:

$$R(\alpha_u(O) \mid O) = \sum_{s \in \mathbf{W}_h} L(u,s) P(s \mid O), \tag{1}$$

where $P(s \mid O)$ is the posterior probability of the word sequence s given that the acoustic vector sequence O is observed. Therefore, the Bayes decision $\alpha_{opt}(O)$ is made by selecting the action with the minimum expected loss, i.e.,

$$\begin{aligned} \alpha_{opt}(O) &= \arg\min_{u \in \mathbf{W}_h} R\left(\alpha_u(O) \mid O\right) \\ &= \arg\min_{u \in \mathbf{W}_h} \sum_{s \in \mathbf{W}_h} L(u,s) P(s \mid O) \end{aligned}. \tag{2}$$

In supervised training, on the other hand, the correct transcription of each training utterance O is known, and the overall risk $\tilde{R}_{all}$ of all possible training utterances is defined as:

$$\tilde{R}_{all} = \int R(\alpha_c(O) \mid O) P(O) dO, \tag{3}$$

where the integral extends over the whole acoustic space. However, in practice, we can only obtain the approximate overall risk R_{all} by summing the risks over a finite number of training utterances, i.e.,

$$\begin{aligned} R_{all} &= \sum_r R(\alpha_{c_r}(O_r) \mid O_r) P(O_r) \\ &= \sum_r \sum_{s \in \mathbf{W}_h^r} L(c_r, s) P(s \mid O_r) P(O_r) \end{aligned}, \tag{4}$$

where $\mathbf{W}_h^r$ and c_r, respectively, denote a set of likely hypothesized word sequences and the reference word sequence associated with the training utterance O_r; and the distribution $P(s \mid O_r)$ is always assumed to be governed by some underlying parametric distributions. To ensure that ASR is as accurate as possible, we need to design a classifier and estimate the parameters in $P(s \mid O_r)$ more carefully in order to minimize the overall risk R_{all}. By applying

the Bayes rule and replacing the probability $P(O_r \mid s)$ with its parameterization, $p_\lambda(O_r \mid s)$, Eq. (4) can be expressed as:

$$R_{all} = \sum_r \frac{\sum_{s \in \mathbf{W}_h^r} L(c_r, s) p_\lambda(O_r \mid s) P(s)}{\sum_{u \in \mathbf{W}_h^r} p_\lambda(O_r \mid u) P(u)} P(O_r), \tag{5}$$

where $p_\lambda(O_r \mid s)$ and $p_\lambda(O_r \mid u)$ are, respectively, the acoustic model likelihoods for s and u under the acoustic model parameter set λ; and $P(s)$ and $P(u)$ are the respective language model probabilities for s and u. The parameters of both the acoustic model and the language model can be estimated by minimizing R_{all}. However, in this study, we only focus on the discriminative estimation of the acoustic model parameters, and adopt the conventional approach for language model training. Moreover, it is assumed that the prior probability $P(O_r)$ is uniformly distributed. As a result, the overall risk becomes

$$R_{all} = \sum_r \frac{\sum_{s \in \mathbf{W}_h^r} L(c_r, s) p_\lambda(O_r \mid s) P(s)}{\sum_{u \in \mathbf{W}_h^r} p_\lambda(O_r \mid u) P(u)}, \tag{6}$$

and the optimal parameter set, λ_{opt}, can be estimated by minimizing the overall risk of the training utterances

$$\lambda_{opt} = \arg\min_\lambda \sum_r \frac{\sum_{s \in \mathbf{W}_h^r} L(c_r, s) p_\lambda(O_r \mid s) P(s)}{\sum_{u \in \mathbf{W}_h^r} p_\lambda(O_r \mid u) P(u)}. \tag{7}$$

To minimize the overall risk, as shown by Equations (4) to (7), the hypothesized word sequence with a lower loss should have a larger posterior probability, and vice versa. How to select an appropriate loss function $L(\cdot,\cdot)$ used in the above equations remains an open research issue. In most pattern classification tasks, to minimize the probability of classification errors, the loss function is often chosen based on the minimum error rate (MER) criterion. This leads directly to the following symmetrical zero-one loss function [Duda *et al.* 2000]:

$$L(u, s) = \begin{cases} 0 & , u = s \\ 1 & , u \neq s \end{cases}. \tag{8}$$

The loss function assigns no loss if $u = s$, and assigns a unit loss when a classification error occurs. In ASR, a hypothesized word sequence that is identical to the correct transcription does not introduce a loss; however, a hypothesized word sequence containing one or more

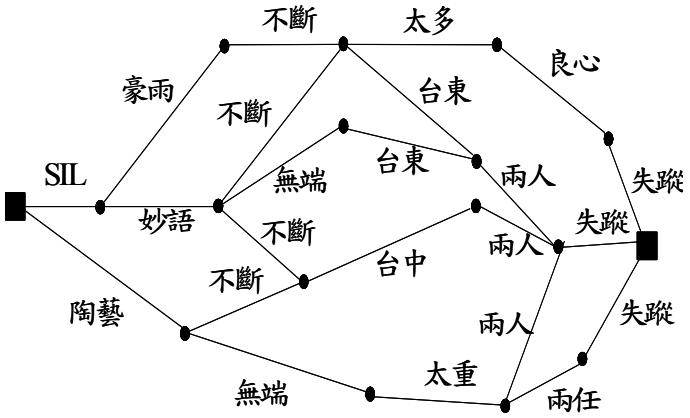


Figure 1. A word lattice can efficiently encode a large number of possible hypothesized word sequences.

word errors, or a totally different sequence, compared to the correct sequence, will incur the same unit loss. Thus, minimizing the overall risk is equivalent to minimizing the expected string error rate (SER) of the training utterances. Nevertheless, SER is not a sufficient metric for the evaluation of ASR performance because, with this metric, all incorrectly hypothesized word sequences are regarded as having the same cost of recognition risk. Instead, the loss function could be defined as the distance of the hypothesized word sequence to the correct transcription. For this purpose, the string edit or Levenshtein distance [Levenshtein 1966] associated with the word error rate (WER) can be adopted. It is believed that WER is more suitable than SER in reflecting differences in ASR results. Optimization using the Levenshtein-based loss function is often referred to as word error minimization (WEM).

However, in complicated ASR tasks, such as LVCSR, it is impossible to perform optimization over the hypothesized space $\mathbf{W}_h^r$ of each training utterance O_r without using a pruning technique because such hypothesized spaces usually contain an extremely large number of hypothesized word sequences. Recently, some practical strategies have been proposed to resolve this problem. For instance, a reduced hypothesized space in the form of an *N*-best list [Schwartz and Chow 1990] or a lattice [Ortmanns 1997] can be generated for each training utterance by only retaining recognized hypotheses with higher probabilities. The optimization process can then be applied efficiently to the reduced hypothesized space. Figure 1 illustrates an example of a word lattice.

3. Minimum Phone Error (MPE) Training

This section describes in detail the application of the minimum phone error (MPE) criterion to acoustic model training. As mentioned in the previous section, the hypothesized space $\mathbf{W}_h^r$ of a given training utterance O_r can be reduced to a smaller space represented by a number of the most likely hypothesized word sequences associated with O_r. The *N*-best list contains the *N* most likely sequences generated by applying the Viterbi algorithm, which has to retain at least *N*-best search hypotheses at both the HMM (Hidden Markov Model) acoustic model-level and word-level recombination points during the speech decoding process. For each hypothesized word sequence on the *N*-best list, it is relatively easy to compute the standard Levenshtein distance to the correct transcription directly. Based on this observation, Kaiser *et al*. proposed overall risk criterion estimation (ORCE) for acoustic model training [Kaiser *et al*. 2000, 2002; Na *et al*. 1995]. This approach takes the *N*-best list as the reduced hypothesized space to obtain training statistics, and applies the extended Baum-Welch algorithm [Gopalakrishnan *et al.* 1991; Normandin 1991] for parameter optimization. In experiments on the TIMIT database, the authors achieved a 21% word error rate reduction compared to the baseline system. However, an *N*-best list usually contains too much redundant information, i.e., two hypothesized word sequences may look very similar, which makes the training procedure inefficient. An alternative representation is the word lattice (or graph), illustrated in Figure 1, which only stores hypothesized word arcs at different segments of the time frames. Although it cannot be guaranteed that all word sequences generated from a word lattice will have higher probabilities than those not presented, it is believed that the approximation will not affect the performance significantly. Nevertheless, for the lattice structure, using the standard Levenshtein distance measure as the loss function is an issue, since it makes the implementation of computing the distance more complicated. Recently, two approaches have been proposed to deal with this problem. One focuses on how to design loss functions that approximate the Levenshtein distance measure, such as MPE training. The other concentrates on the design of algorithms to segment the word lattice so as to make the computation of the Levenshtein distance feasible, such as the minimum Bayes risk discriminative training (MBRDT) approach [Doumpiotis *et al.* 2003, 2004]. To efficiently reduce the complexity of the hypothesized space in MBRDT, a lattice segmentation algorithm is applied to divide the lattice into several non-overlapping components. It has been shown that MBRDT achieves a considerable performance improvement over the baseline system trained with the maximum likelihood (ML) criterion.

The MPE training approach, which is one of the most attractive discriminative training techniques, tries to optimize an acoustic model's parameters by minimizing the expected phone error rate. The objective function of MPE is given as [Povey 2004]:

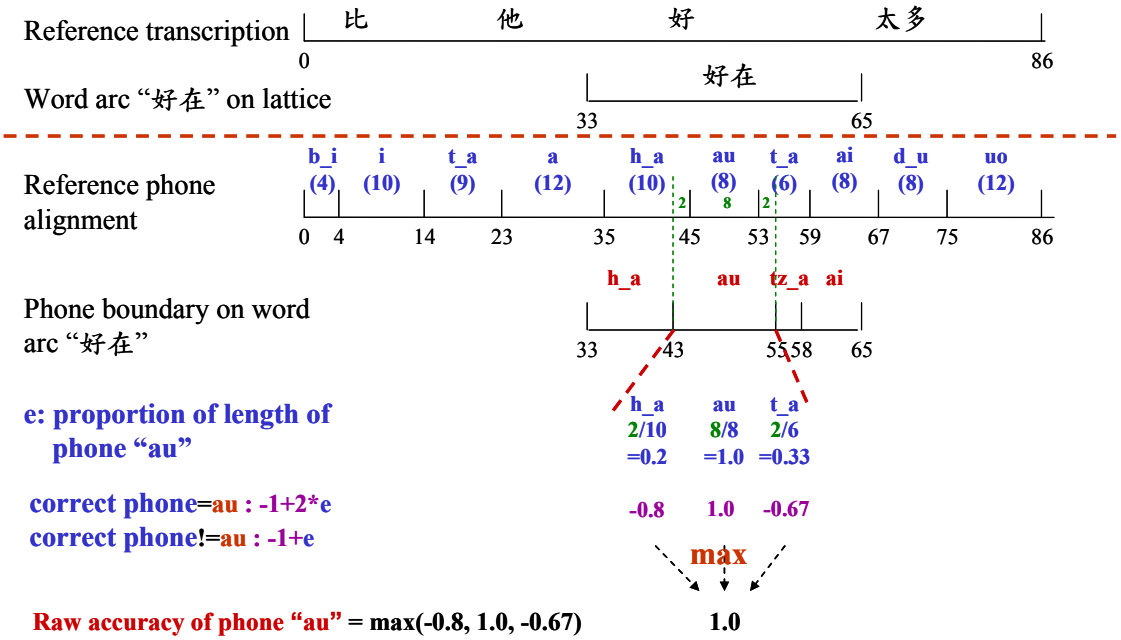


Figure 2. Raw phone accuracy calculation.

$$F_{MPE}(\lambda) = \sum_{r} \frac{\sum_{s \in \mathbf{W}_{lat}^{r}} p_{\lambda}(O_r \mid s) P(s) A(c_r, s)}{\sum_{u \in \mathbf{W}_{lat}^{r}} p_{\lambda}(O_r \mid u) P(u)}, \tag{9}$$

where $\mathbf{W}_{lat}^{r}$ is the lattice generated by the speech recognizer, used to represent a reduced hypothesized space of word sequences; and $A(c_r, s)$ is the raw accuracy of word sequence s, which is an approximation of the true accuracy computed globally using the standard Levenshtein distance. It is obvious that maximizing the objective function is equivalent to minimizing the expected phone error. The raw accuracy $A(c_r, s)$ is defined as:

$$A(c_r, s) = \sum_{q \in s} A'(c_r, q), \tag{10}$$

where q is the phone involved in s, and $A'(c_r, q)$ is a local function used to calculate the raw phone accuracy of each phone q in s. The phone accuracy is calculated locally on each phone arc of the word lattice, instead of globally on each hypothesized word sequence. Given a word arc on the word lattice, the time boundaries of the phone arcs can be determined by aligning the corresponding speech segment with its constituent HMM acoustic models. Figure 2 shows the calculation of raw phone accuracy. Notice that we adopt INITIAL/FINAL units instead of phone units as the acoustic units in our Mandarin LVCSR system. Therefore, for

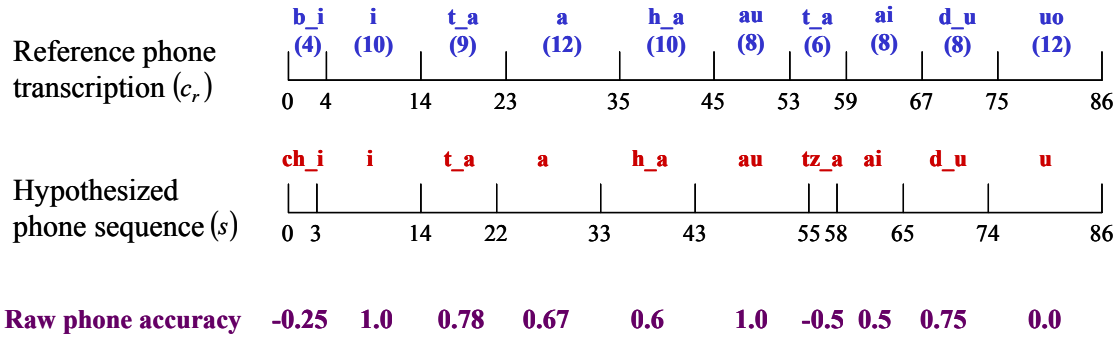


Figure 3. Approximate accuracy versus exact accuracy.

simplicity, each INITIAL or FINAL unit is regarded as a phone in the elucidation. In Figure 2, the raw phone accuracy of phone *"au"* involved in the word arc "好在" is calculated in the following steps. First, the word arc "好在" is aligned with time boundaries of a phone sequence to obtain the start and end time boundaries of the phone *"au"*. Second, for each phone q' in the correct transcription, we calculate the overlapped portion of *"au"* in time frames, and denote it as $e(q',"au")$. Finally, the raw phone accuracy of phone *"au"*, i.e., $A'(c_r,"au")$, is calculated using the following formula:

$$A'(c_r,"au") = \max_{q'} \begin{cases} -1+2e("au",q') & \text{if } q'="au" \\ -1+e("au",q') & \text{otherwise} \end{cases}. \tag{11}$$

It is obvious that $A'(c_r,"au")$ ranges from 1 to $-1+1/T_r$, where T_r is the length of observation O_r in terms of the time frames. For example, if the phone arc *"au"* overlays at least one phone q' in the correct transcription with the same identity in time, *"au"* is considered to be a correct phone, i.e., $A'(c_r,"au")=1$. Figure 3 compares the accuracy of a hypothesized word sequence obtained via the approximate function discussed here and the exact calculation using the Levenshtein distance.

According to Povey's work [Povey 2004], the auxiliary function for optimizing the objective function of MPE in Eq. (9) is

$$H_{MPE}(\lambda,\bar{\lambda}) = \sum_r \sum_{q\in \mathbf{W}_{lat}^r} \left.\frac{\partial F_{MPE}(\lambda)}{\partial \log p_\lambda(O_r \mid q)}\right|_{\lambda=\bar{\lambda}} \log p(O_r \mid q), \tag{12}$$

where $\bar{\lambda}$ is the current model parameter set, q is a specific phone arc in $\mathbf{W}_{lat}^{r}$, and $p_{\lambda}(O_r \mid q)$ is the likelihood given the phone arc q. Note that $H_{MPE}(\lambda, \bar{\lambda})$ is a weak-sense auxiliary function of $F_{MPE}(\lambda)$ around $\lambda = \bar{\lambda}$ with the following property:

$$\left.\frac{\partial F_{MPE}(\lambda)}{\partial \lambda}\right|_{\lambda=\bar{\lambda}} = \left.\frac{\partial H_{MPE}(\lambda, \bar{\lambda})}{\partial \lambda}\right|_{\lambda=\bar{\lambda}}. \tag{13}$$

In other words, both the objective and auxiliary functions have the same derivative with respect to λ when they are evaluated at the current estimate $\bar{\lambda}$. For simplicity, we only consider the MPE-based estimation of mean vectors and covariance matrices in HMMs. The state transition probabilities and mixture weights trained by the ML criterion remain unchanged. As a result, in this study, the final auxiliary function for MPE training is expressed as:

$$g_{MPE}(\lambda, \bar{\lambda}) = \sum_r \sum_{q \in \mathbf{W}_{lat}^{r}} \sum_{t=s_q}^{t=e_q} \sum_{m} \gamma_q^{rMPE} \gamma_{qm}^{r}(t) \log N(o_r(t), \mu_m, \Sigma_m), \tag{14}$$

where s_q and e_q represent the start and end times of the phone arc q, respectively; m is the mixture index of the acoustic models; μ_m and Σ_m are, respectively, the mean vector and covariance matrix for mixture m; $\gamma_{qm}^{r}(t)$ is the occupation probability for mixture m on q; $o_r(t)$ is the observation vector at time t; and γ_q^{rMPE} represents $\left.\frac{\partial F_{MPE}(\lambda)}{\partial \log p_{\lambda}(O_r \mid q)}\right|_{\lambda=\bar{\lambda}}$ in Eq. (12), which can be expressed as:

$$\begin{aligned}\left.\frac{\partial F_{MPE}(\lambda)}{\partial \log p_{\lambda}(O_r \mid q)}\right|_{\lambda=\bar{\lambda}} = & \frac{\sum_{v' \in \mathbf{W}_{lat}^{r}, q \in v'} p_{\bar{\lambda}}^{-}(O_r \mid v') P(v') A(v', s_r)}{\sum_{u' \in \mathbf{W}_{lat}^{r}, q \in u'} p_{\bar{\lambda}}^{-}(O_r \mid u') P(u')} \frac{\sum_{u' \in \mathbf{W}_{lat}^{r}, q \in u'} p_{\bar{\lambda}}^{-}(O_r \mid u') P(u')}{\sum_{u \in \mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}^{-}(O_r \mid u) P(u)} \\ & - \frac{\sum_{v \in \mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}^{-}(O_r \mid v) P(v) A(v, s_r)}{\sum_{u \in \mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}^{-}(O_r \mid u) P(u)} \frac{\sum_{u' \in \mathbf{W}_{lat}^{r}, q \in u'} p_{\bar{\lambda}}^{-}(O_r \mid u') P(u')}{\sum_{u \in \mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}^{-}(O_r \mid u) P(u)}\end{aligned}. \tag{15}$$

In Eq. (15), $\frac{\sum_{u' \in \mathbf{W}_{lat}^{r}, q \in u'} p_{\bar{\lambda}}^{-}(O_r \mid u') P(u')}{\sum_{u \in \mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}^{-}(O_r \mid u) P(u)}$ is the occupation probability of phone arc q; $\frac{\sum_{v' \in \mathbf{W}_{lat}^{r}, q \in v'} p_{\bar{\lambda}}^{-}(O_r \mid v') P(v') A(v', s_r)}{\sum_{u' \in \mathbf{W}_{lat}^{r}, q \in u'} p_{\bar{\lambda}}^{-}(O_r \mid u') P(u')}$ is the weighted average accuracy of hypothesized word sequences

in $\mathbf{W}_{lat}^{r}$ that include q; and $\frac{\sum_{v\in\mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}(O_r \mid v)P(v)A(v,s_r)}{\sum_{u\in\mathbf{W}_{lat}^{r}} p_{\bar{\lambda}}(O_r \mid u)P(u)}$ is the weighted average accuracy of all hypothesized word sequences in $\mathbf{W}_{lat}^{r}$. All three quantities can be calculated efficiently.

Since maximizing the weak sense auxiliary function with respect to λ does not guarantee an increase in the objective function, the auxiliary function is augmented with an extra smoothing function $g_{EB}^{smooth}(\lambda,\bar{\lambda})$ to moderate the parameter update and prevent extreme parameter values being estimated. The following is an example of a smoothing function:

$$g_{EB}^{smooth}(\lambda,\bar{\lambda}) = \sum_{m} -\frac{D_m}{2}\left[\log(|\Sigma_m|) + (\mu_m-\bar{\mu}_m)^T\Sigma_m^{-1}(\mu_m-\bar{\mu}_m) + tr(\bar{\Sigma}_m\Sigma_m^{-1})\right], \tag{16}$$

where D_m is a per-mixture level controlling constant. Note that $g_{EB}^{smooth}(\lambda,\bar{\lambda})$ is deemed a log-Gaussian prior distribution with a differential value of zero with respect to λ when it is evaluated at the current estimate $\bar{\lambda}$. Therefore, the differentials of the augmented auxiliary function with respect to μ_m and Σ_m are computed as shown, respectively, in the following equations:

$$\frac{\partial(g_{MPE}(\lambda,\bar{\lambda}) + g_{EB}^{smmoth}(\lambda,\bar{\lambda}))}{\partial\mu_m} = \sum_r \sum_{q\in\mathbf{W}_{lat}^{r}} \sum_{t=s_q}^{t=e_q} \gamma_q^{rMPE}\gamma_{qm}^{r}(t)\Sigma_m^{-1}\left(o_r(t)-\mu_m\right) - D_m\left[\Sigma_m^{-1}(\mu_m-\bar{\mu}_m)\right], \tag{17}$$

$$\frac{\partial(g_{MPE}(\lambda,\bar{\lambda}) + g_{EB}^{smooth}(\lambda,\bar{\lambda}))}{\partial\Sigma_m^{-1}} = \sum_r \sum_{q\in\mathbf{W}_{lat}^{r}} \sum_{t=s_q}^{t=e_q} \gamma_q^{rMPE}\gamma_{qm}^{r}(t)\left[\frac{1}{2}\Sigma_m^T - \frac{1}{2}\left(\left(o_r(t)-\mu_m\right)\left(o_r(t)-\mu_m\right)^T\right)\right]. + \frac{D_m}{2}\left[\Sigma_m^T - (\mu_m-\bar{\mu}_m)(\mu_m-\bar{\mu}_m)^T - \bar{\Sigma}_m^T\right] \tag{18}$$

Next, by completing the differentiations and equating the above equations to zero, the following Extended Baum-Welch (EB) update formulae [Normandin 1991] are derived:

$$\mu_m = \frac{\sum_r \sum_{q\in\mathbf{W}_{lat}^{r}} \sum_{t=s_q}^{t=e_q} \gamma_q^{rMPE}\gamma_{qm}^{r}(t)o_r(t) + D_m\bar{\mu}_m}{\sum_r \sum_{q\in\mathbf{W}_{lat}^{r}} \sum_{t=s_q}^{t=e_q} \gamma_q^{rMPE}\gamma_{qm}^{r}(t) + D_m}, \tag{19}$$

$$\Sigma_m = \frac{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) o_r(t) o_r(t)^T + D_m \left[\overline{\Sigma}_m + \overline{\mu}_m \overline{\mu}_m^T \right]}{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) + D_m} - \mu_m \mu_m^T . \tag{20}$$

Moreover, to incorporate the ML estimate and smooth the update, the so-called I-smoothing technique [Povey and Woodland 2002] is employed to provide a better estimate. I-smoothing is also regarded as a prior distribution for smoothing the auxiliary function, where the mode of the distribution is the same as the estimate obtained by ML training. The update equations thus become:

$$\mu_m = \frac{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) o_r(t) + D_m \overline{\mu}_m + \frac{\tau_m}{\gamma_m^{ML}} \theta_m^{ML}(O)}{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) + D_m + \tau_m}, \tag{21}$$

$$\Sigma_m = \frac{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) o_r(t) o_r(t)^T + D_m \left[\overline{\Sigma}_m + \overline{\mu}_m \overline{\mu}_m^T \right] + \frac{\tau_m}{\gamma_m^{ML}} \theta_m^{ML}(O^2)}{\sum_r \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{r\,MPE} \gamma_{qm}^r(t) + D_m + \tau_m} - \mu_m \mu_m^T , \tag{22}$$

where τ_m is a constant, and γ_m^{ML}, $\theta_m^{ML}(O)$, and $\theta_m^{ML}(O^2)$ are further expressed, respectively, as:

$$\gamma_m^{ML} = \sum_r \sum_t \gamma_m^{r\,ML}(t), \tag{23}$$

$$\theta_m^{ML}(O) = \sum_r \sum_t \gamma_m^{r\,ML}(t) o_r(t), \tag{24}$$

and

$$\theta_m^{ML}(O^2) = \sum_r \sum_t \gamma_m^{r\,ML}(t) o_r(t) o_r(t)^T . \tag{25}$$

In each of the above equations, $\gamma_m^{r\,ML}(t)$ is the ML occupation probability for mixture m. I-smoothing can also be considered as an interpolation between the MPE estimate and the ML estimate. As $\tau_m \to \infty$, it performs like ML training. On the other hand, it behaves purely as MPE training when $\tau_m \to 0$. Basically, the technique provides better results when the value of τ_m is properly chosen (e.g., we adopted a setting of $\tau_m = 10$ in our experiments). Recently, it has been verified that using the statistics of MMI (Maximum Mutual Information) training in I-smoothing can further improve the estimate [Zheng and Stolcke 2005; Povey *et al.* 2005].

Finally, let us examine the quantity $\gamma_q^{r\,MPE}$ in more detail. To simplify the discussion, we adopt the following equations:

$$\gamma_q^r = \frac{\sum_{u' \in \mathbf{W}_{lat}^r, q \in u'} p_{\bar{\lambda}}(O_r \mid u') P(u')}{\sum_{u \in \mathbf{W}_{lat}^r} p_{\bar{\lambda}}(O_r \mid u) P(u)}, \tag{26}$$

$$c^r(q) = \frac{\sum_{v' \in \mathbf{W}_{lat}^r, q \in v'} p_{\bar{\lambda}}(O_r \mid v') P(v') A(v', s_r)}{\sum_{u' \in \mathbf{W}_{lat}^r, q \in u'} p_{\bar{\lambda}}(O_r \mid u') P(u')}, \tag{27}$$

$$c_{avg}^r = \frac{\sum_{v \in \mathbf{W}_{lat}^r} p_{\bar{\lambda}}(O_r \mid v) P(v) A(v, s_r)}{\sum_{u \in \mathbf{W}_{lat}^r} p_{\bar{\lambda}}(O_r \mid u) P(u)}, \tag{28}$$

where $c^r(q)$ is the weighted average phone accuracy of hypothesized word sequences that involve q; and c_{avg}^r is the weighted average phone accuracy of all hypothesized word sequences in $\mathbf{W}_{lat}^r$. It is clear that the three main statistics must be gathered by applying the forward-backward algorithm to the word lattice [Povey 2004]. Note that the term $c^r(q) - c_{avg}^r$ reflects the difference in the weighted average phone accuracy between the word sequences containing arc q and all word sequences in the lattice. As $c^r(q) = c_{avg}^r$, no training statistics are contributed to phone arc q in MPE training. Positive contributions are made to arc q if $c^r(q)$ is greater than c_{avg}^r, i.e., if phone arc q is more accurate than the average. Conversely, if $c^r(q)$ is smaller than c_{avg}^r, negative contributions are made to arc q and thus show the discrimination. For a reasonable combination of acoustic model likelihoods and language model probabilities, it is necessary to restrict the acoustic likelihoods by introducing an exponential scaling factor. The scaling factor is empirically set depending on the task at hand; in our experiments, we adopted a value of 1/12. Alternatively, a word unigram language model constraint can be used to improve the generalization capabilities of such discriminative

training.

4. MPE-based Linear Regression (MPELR) Adaptation

Acoustic model adaptation, which is one of the most important topics in ASR, tries to eliminate some of the spoken and environmental variations between the training and test sets. However, it is a challenging task to adjust the large number of acoustic model parameters when only a very small amount of data is available for model adaptation. To ensure a more reliable estimation of acoustic model parameters, transformation-based approaches have been developed to adapt the acoustic model indirectly by using a set of affine transforms, such as the maximum likelihood linear regression (MLLR) adaptation [Leggetter and Woodland 1995]. Similarly, word or phone error minimization approaches can be used to estimate the transformation matrices. Among these approaches, we focus on MPE-based linear regression (MPELR) adaptation [Wang and Woodland 2004], which obtains the transformation matrices by using the MPE criterion.

As in typical MLLR adaptation, Gaussian components are first clustered into several regression classes. Components in the same class share the same transformation matrix. The Gaussian mean vectors are transformed by:

$$\mu_m = A_k \bar{\mu}_m + b_k = W_k \bar{\xi}_m , \tag{29}$$

where the subscript k is the class index; $W_k = \left[b_k \ A_k \right]$ is a $d \times (d+1)$ transformation matrix; and $\bar{\xi}_m = \left[1 \ \bar{\mu}_m^T \right]^T$ is the $(d+1)$-dimensional extended mean vector based on the current estimate. Meanwhile, the covariance matrices can be updated by [Gales and Woodland 1996]

$$\Sigma_m = \bar{L}_m^{-T} H_k \bar{L}_m^{-1} , \tag{30}$$

where H_k is the linear transformation matrix to be estimated for the class k, and $\bar{L}_m$ is the Cholesky factor of $\bar{\Sigma}_m^{-1}$. Hereafter, for simplicity, the subscript k representing the cluster index is omitted. Based on Eq. (14), the auxiliary function can be derived as:

$$g_{MPE}(\{W,H\},\{\bar{W},\bar{H}\}) = \sum_m \sum_{q \in \mathbf{W}_{lat}^r} \sum_{t=s_q}^{t=e_q} \gamma_q^{MPE} \gamma_{qm}(t) \log N(o(t); W\xi_m, L_m^{-T} H L_m^{-1}) . \tag{31}$$

Like MPE training, described in Section 3, the auxiliary function in Eq. (31) can be further augmented with an extra smoothing function $g_{EBW}^{smooth}(\{W,H\},\{\bar{W},\bar{H}\})$ to derive a more reliable estimation of the transformation matrices. This is usually given by:

$$
\begin{aligned}
& g_{EBW}^{smooth}(\{W,H\},\{\bar{W},\bar{H}\}) \\
&= \sum_m -\frac{D_m}{2}\Big[\log\Big(|\, L_m^{-T} H L_m^{-1}\,|\Big) + (W\xi_m - \bar{W}\xi_m)^T L_m H^{-1} L_m^T (\bar{W}\xi_m - W\xi_m)\,, \\
& \quad + tr\Big(L_m^{-T}\bar{H}L_m^{-1} L_m H^{-1} L_m^T\Big)\Big]
\end{aligned}
\tag{32}
$$

where $tr(\cdot)$ is the standard matrix trace operation. After differentiating the auxiliary function with respect to W and setting it to zero, we get the following closed-form solution:

$$
\begin{aligned}
& \sum_m \Sigma_m^{-1}\left(\sum_{q\in\mathbf{W}_{lat}^r}\sum_{t=s_q}^{t=e_q}\gamma_q^{MPE}\gamma_{qm}(t)o(t) + D_m\bar{W}\xi_m\right)\xi_m^T \\
&= \sum_m \left(\sum_{q\in\mathbf{W}_{lat}^r}\sum_{t=s_q}^{t=e_q}\gamma_q^{MPE}\gamma_{qm}(t) + D_m\right)\Sigma_m^{-1} W\xi_m\xi_m^T
\end{aligned}
\tag{33}
$$

The above equation can be solved row-by-row using the Gaussian elimination method to obtain the re-estimation formula for the transformation matrix of mean vectors. The re-estimation formula for the transformation matrix of covariance matrices can be derived in a similar way.

Again, to improve the generalization of the test set, extra prior information, such as the ML statistics, can be considered. Therefore, the final auxiliary function employed in this paper is augmented with the following smoothing function:

$$
g^{I-smooth}(W,H) = \sum_m \frac{\tau_m}{\gamma_m^{ML}}\sum_t \gamma_m{}^{ML}(t)\log N(o(t);W\xi_m, L_m^{-T}HL_m^{-1})\,. \tag{34}
$$

5. Word Error Minimization (WEM) Decoding

Given a speech utterance, the standard maximum a posteriori (MAP) decoding approach tries to output the hypothesized word sequence with the highest posterior probability. Actually, by substituting a zero-one loss function into Eq. (2), the MAP decoding formula can be derived. This implies that the MAP decoding approach is based on minimizing the string error rate (SER). Thus, it only provides suboptimal results when the ASR performance is measured in terms of the word error rate (WER) or the character error rate (CER). Hence, replacing the zero-one loss function in Eq. (2) with the Levenshtein distance measure leads to the WEM decoding approach, which finds the hypothesized word sequence with the minimum WER or CER. However, as mentioned in Section 3, a direct implementation of WEM decoding with the word lattice is complicated because there is still no efficient algorithm for computing the

Table 1. Detailed statistics of the training and test sets.

Gender	Training set			Test set			#Speakers in the training and test sets
	Total length (sec)	Total Syllables	#Speakers	Total length (sec)	Total Syllables	#Speakers	
Male	46,001.3	545,732	≤66	1,301.4	26,219	9	9
Female	46,007.2		≤111	3,914.0		≤23	≥13

Levenshtein distance between any two possible word sequences in the word lattice. To make the implementation of the WEM decoding approach feasible, we initially employ an *N*-best list of hypothesized word sequences. The WEM decoding approach can then be applied explicitly by choosing the hypothesized word sequence with the minimum expected risk [Stolcke *et al.* 1997]. The decision formula can thus be expressed as:

$$\alpha_{opt}(O) = \arg\min_{u \in N-\text{Nest}} \sum_{s \in N-\text{Nest}} \frac{p\ (O \mid s)p(s)}{\sum_{v \in N-\text{Nest}} p(O \mid v)p(v)} L(u,s)\,, \tag{35}$$

where u, s, and v are hypothesized word sequences in the *N*-best list. Similar ideas have been proposed recently by Mangu *et al.* [Mangu *et al.* 2000] and Goel and Byrne [Goel and Byrne 2000]. As an alternative, a novel optimal Bayes decision (OBC) approach for word lattice rescoring has been developed [Chien *et al.* 2006]. It also provides a promising framework for WEM decoding.

6. Experiment Setup

In this section, we describe the large vocabulary continuous speech recognition system and the speech and text data used in this paper.

6.1 Front-End Signal Processing

Front-end processing was performed with the HLDA-based (Heteroscedastic Linear Discriminant Analysis) data-driven Mel-frequency feature extraction approach, and then processed by MLLT (Maximum Likelihood Linear Transformation) transformation for feature de-correlation. In addition, utterance-based feature mean subtraction and variance normalization were applied to all the training and test materials.

6.2 Speech Corpus and Acoustic Model Training

The speech corpus consisted of approximately 198 hours of MATBN (Mandarin Across Taiwan Broadcast News) Mandarin television news content [Wang *et al.* 2005], which was collected by Academia Sinica and the Public Television Service Foundation of Taiwan between November 2001 and April 2003. All the speech materials were manually segmented into separate stories, each of which was spoken by one news anchor, several field reporters, and interviewees. Some stories contained background noise, speech, and music. All 198 hours of speech data was accompanied by corresponding orthographic transcripts, of which about 25 hours of gender-balanced speech data of the field reporters collected from November 2001 to December 2002 was used to bootstrap the acoustic training. The training set consisted of 545,732 syllables and the average length of a word was 1.65 characters. Another set of data, 1.5 hours in length, collected during 2003 was reserved for testing. Due to the limited number of distinct field reporters in the corpus, some test data belonged to the training field reporters. The test set consisted of 26,219 syllables and the average word length was also 1.65 characters. Table 1 shows the detailed statistics of the training and test sets.

The acoustic models chosen for speech recognition were a silence model, 112 right-context-dependent INITIAL models, and 38 context-independent FINAL models. Each INITIAL model was represented by an HMM with 3 states, while each FINAL model had 4 states. Note that gender-independent models were used. The Gaussian mixture number per state ranged from 2 to 128, depending on the amount of training data. The acoustic models were first trained using the ML criterion and the Baum-Welch updating formulae. The MPE-based and MMI (Maximum Mutual Information)-based [Povey and Woodland 2002] acoustic model training approaches were further applied to acoustic models pre-trained by the ML criterion. Unigram language model constraints were used to collect the training statistics from the word lattices for these two training approaches. For MPE training, both silence and short-pause labels were involved in the calculation of the raw phone accuracy of the hypothesized word sequences.

6.3 Lexicon and N-gram Language Modeling

Initially, the recognition lexicon consisted of 67K words. A set of about 5K compound words was automatically derived using forward and backward bigram statistics and added to the lexicon to form a new lexicon of 72K words. The background language models used in this experiment were trigram and bigram models, which were estimated according to the ML criterion using a text corpus consisting of 170 million Chinese characters collected from the Central News Agency (CNA) in 2001 and 2002 (the Chinese Gigaword Corpus released by LDC). The *N*-gram language models were trained with Katz back-off smoothing technique using the SRI Language Modeling Toolkit (SRILM) [Stolcke 2000].

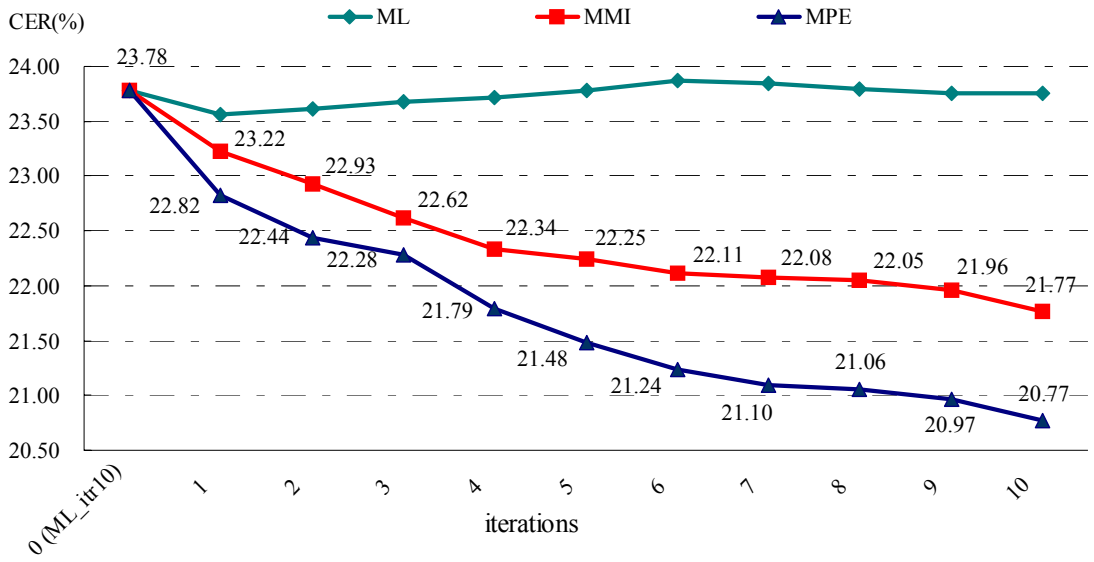


Figure 4. Recognition results, in terms of the CER, for three systems trained on ML, MMI, and MPE criteria, respectively.

6.4 Speech Recognition

The speech recognizer was implemented with a left-to-right frame-synchronous Viterbi tree-copy search and a lexical prefix tree of the lexicon. For each speech frame, a beam pruning technique, which considered the decoding scores of path hypotheses together with their corresponding unigram language model look-ahead scores and syllable-level acoustic look-ahead scores [Chen *et al.* 2005], was used to select the most promising path hypotheses. Moreover, if the word hypotheses ending at each speech frame had higher scores than a predefined threshold, their associated decoding information, such as the word start and end frames, the identities of current and predecessor words, and the acoustic score, were kept to build a word lattice for further language model rescoring. We used the word bigram language model in the tree search procedure and the trigram language model in the word lattice rescoring procedure.

7. Experiment Results and Discussions

Now, a series of experiments performed to assess speech recognition as a function of the acoustic training and adaptation approaches, as well as the speech decoding approaches will be presented.

Table 2. Recognition results of the acoustic model training and unsupervised adaptation approaches

	INITIAL/FINAL Error Rate (%)	Character Error Rate (%)
ML	13.56	23.78
(ML+) MPE	11.12	20.77
(ML+) MPE + MLLR	10.94	20.45
(ML+) MPE + MPELR	10.82	20.29

7.1 Experiments on MPE Acoustic Model Training

The acoustic models of the baseline system were first trained using the ML criterion with 10 iterations of Baum-Welch updating. Then, MPE training (with an optimum setting of $\tau_m = 10$) was applied to the ML-trained acoustic models. In the implementation, we calculated the raw accuracy of each INITIAL/FINAL, instead of each phone, i.e., we had actually performed Minimum INITIAL/FINAL Error training, not Minimum Phone Error training, in the Mandarin LVCSR system. While evaluating the ASR performance, neither the silence nor the short-pause labels were included in the calculation of CER. MMI training was also performed for comparison with MPE training. As mentioned previously, for both MPE and MMI training, unigram language model constraints were imposed when collecting the training statistics from the word lattices. The results for acoustic model training are shown in Figure 4. We observe that the ML-trained baseline system (at the 10th iteration) yields a CER of 23.78%. On the other hand, both MMI and MPE work very well, providing a great boost to the acoustic models initially trained by ML. The acoustic models trained by MPE consistently outperform those trained by MMI across all training iterations. In summary, the MPE-trained acoustic models achieve a relative CER reduction of 12.66% (at the 10th iteration) over those trained by ML. Moreover, as shown in Table 2, the improvements are consistent. The INITIAL/FINAL model error rate is reduced from 13.56% (baseline, ML training only) to 11.12% (at the 10th MPE training iteration). The 18% relative error rate reduction demonstrates the effectiveness of the Minimum INITIAL/FINIAL Error training approach, and the improvement in the acoustic models leads to a 3% absolute reduction in CER (from 23.78% to 20.77%). The use of statistical linguistic rules in MPE training still plays an important role in re-weighting the occupancy statistics, especially in an LVCSR system. In our previous work [Kuo 2005], it was found that much of the CER improvement was lost without embedding the language weight.

The question thus arises: What makes MPE superior to MMI? In Eq. (7), if the summation operator over all training utterances is replaced by the product operator and the loss function is the zero-one function in Eq. (8), one gets the following MMI criterion:

$$\lambda_{MMI} = \arg\max_{\lambda} \sum_{r} \log \frac{p_{\lambda}(O_r \mid s)p(s)}{\sum_{u \in \mathbf{W}_h^r} p_{\lambda}(O_r \mid u)p(u)}, \tag{36}$$

which maximizes the logarithmic product of the posterior probabilities of the reference transcriptions. The use of the zero-one loss function implies that MMI tends to minimize the sentence error rate. Hence, it is reasonable to say that MMI is inferior to MPE in terms of CER.

7.2 Experiments on Unsupervised MPELR Acoustic Model Adaptation

In this subsection, we evaluate the performance of the MPE-based unsupervised acoustic model adaptation approach. In these experiments, utterance-based unsupervised adaptation was used. First, each test utterance was decoded using the MPE-trained acoustic models. Then, after the forward-backward stage to gather sufficient statistics, the acoustic models were adapted according to the recognized transcriptions. All the Gaussian components of the HMM acoustic models were clustered into three broad phonetic regression classes (i.e., INITIAL, FINAL, and Silence) in advance. Only the mean vectors of each Gaussian component were adapted because it has been found that adapting the mean vectors alone yields the most improvement [Gales and Woodland 1996]. Unsupervised MLLR adaptation was performed as the baseline. In the experiment results presented in Table 2, comparing Row 4 (MPE + MLLR) to Row 3 (MPE), we observe that the CER can be reduced from 20.77% to 20.45%, which indicates that MLLR adaptation can, to some extent, effectively mitigate the degradation of ASR performance caused by different acoustic variations. Row 5 of Table 2 gives the error rate obtained by MPELR adaptation. This result, 0.16% improvement in terms of CER, shows that MPELR is slightly better than MLLR. One possible reason for the insignificant improvement over MLLR is the use of a weak-sense auxiliary function. As a result, the convergence speed of MPE-based techniques is not as fast as the strong-sense auxiliary function used in ML-based techniques. In contrast, the advantage of MPE is that it tries to achieve a lower error rate when over-training is encountered. This is why MPE training is performed after ML training and not for bootstrapping the initial models. Similarly, MPELR adaptation can be performed after MLLR adaptation. However repeated on-line adaptation causes the decoding phase to become tardy, which is why it is only performed once in the online stage.

Table 3. Recognition results (CERs) for N-best list WEM rescoring.

	CER (%)
MPE + MPELR	20.29
MPE + MPELR + WEM	20.23
50-best Error Rate	17.82
Lattice Error Rate	10.12

7.3 Experiments on WEM Decoding

For each test utterance, an *N*-best list of hypothesized word sequences was first generated from the word lattice. We limited the number of hypothesized word sequences included in the *N*-best list to 50, and the Levenshtein distance was calculated in terms of character units. The experiment results are shown in Table 3. From Row 3 (MPE + MPELR + WEM), one observes that, with the best set of acoustic models, WEM only achieves a slight reduction of 0.06% in CER compared to that obtained by conventional MAP decoding, as shown in Row 2. Row 5 (Lattice Error Rate) provides the information regarding the lattice error rate [Ortmanns *et al.* 1997], which is the best achievable lower boundary, by rescoring on the current word lattice. This can be computed by finding the best hypothesized word sequence with the minimum Levenshtein distance to the reference transcription from the corresponding word lattice. On the other hand, Row 4 (50-best Error Rate) gives the lower boundary of the best character error rate for the top 50 hypotheses with the highest scores, which is the true best achievable lower bound in our implementation. From the experiment results, the WEM algorithm seems to achieve an almost imperceptible improvement of about 0.06%. The most likely explanation is that there is a defect in the approximation of the posterior distribution. In addition, the WEM algorithm decides the word sequence with the highest posterior probability in most situations [Schlüter *et al*. 2005]. For the above reasons, we consider that the improvement in CER accuracy is insignificant.

8. Conclusions

In this paper, we have investigated the following word error minimization approaches for Mandarin large vocabulary continuous speech recognition: 1) the MPE criterion used in acoustic model training and adaptation; and 2) the WEM criterion in speech decoding. Unlike conventional techniques, these two approaches try to minimize the expected word error, rather than the string-level error. Experiments on the MATBN corpus demonstrate that MPE training can significantly improve a system initially trained with the ML criterion. Likewise, MPELR adaptation can significantly reduce the CER for the unsupervised adaptation task. This result is superior to that obtained by conventional MLLR adaptation. Finally, *N*-best rescoring using the WEM criterion achieves a slight improvement over traditional MAP decoding. We are

currently conducting an in-depth investigation of the WEM approaches to language modeling [Kuo and Chen, 2005], as well as their comparison and integration with other approaches.

References

Chen, B., J.-W. Kuo, W.-H. Tsai, "Lightly Supervised and Data-Driven Approaches to Mandarin Broadcast News Transcription," *International Journal of Computational Linguistics and Chinese Language Processing*, 10(1), 2005, pp.1-18.

Chien, J.-T., C.-H. Huang, K. Shinoda and S. Furui, "Towards Optimal Bayes Decision for Speech Recognition," in *Proc. ICASSP'06,* 2006.

Doumpiotis, V., S. Tsakalidis and W. Byrne, "Discriminative Training for Segmental Minimum Bayes Risk Decoding," in *Proc. ICASSP'03,* 2003.

Doumpiotis, V., S. Tsakalidis and W. Byrne, "Lattice Segmentation and Minimum Bayes Risk Discriminative Training," in *Proc. Eurospeech'03,* 2003.

Doumpiotis, V. and W. Byrne, "Pinched Lattice Minimum Bayes Risk Discriminative Training for Large Vocabulary Continuous Speech Recognition," in *Proc. ICSLP'04,* 2004.

Duda, R. O., P. E. Hart and D. G. Stork, *Pattern Classification,* 2nd ed. New York: John and Wiley, 2000.

Gales, M. J. F. and P. C. Woodland, "Mean and Variance Adaptation within the MLLR Framework," *Computer Speech and Language,* 10, 1996, pp.249-264.

Goel, V. and W. Byrne, "Minimum Bayes-Risk Automatic Speech Recognition," *Computer Speech and Language*, 14, 2000, pp.115-135.

Gopalakrishnan, P. S., D. Kanevsky, A. Nádas and D. Nahamoo, "An Inequality for Rational Functions with Applications to Some Statistical Estimation Problems," *IEEE Trans. Information Theory,* 37, 1991, pp.107-113.

Kaiser, J., B. Horvat and Z. Kacic, "A Novel Loss Function for the Overall Risk Criterion Based Discriminative Training of HMM Models," in *Proc. ICSLP'00,* 2000.

Kaiser, J., B. Horvat and Z. Kacic, "Overall Risk Criterion Estimation of Hidden Markov Model Parameters," *Speech Communication*, 38, 2000, pp.383-398.

Kuo, J.-W. and B. Chen, "Minimum Word Error Based Discriminative Training of Language Models," in *Proc. INTERSPEECH'05,* 2005.

Kuo, J.-W, "An Initial Study on Minimum Phone Error Discriminative Learning of Acoustic Models for Mandarin Large Vocabulary Continuous Speech Recognition," *Master Thesis*, *National Taiwan Normal University*, June 2005.

Leggetter, C. J. and P. C. Woodland, "Maximum Likelihood Linear Regression for Speaker Adaptation of Continuous Density Hidden Markov Models," *Computer Speech and Language*, 9, 1995, pp.171-185.

Levenshtein, A., "Binary Codes Capable of Correcting Deletions, Insertions and Reversals," *Soviet Physics Doklady*, 10(8), 1966, pp.707-710.

Mangu, L., E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language,* 14, 2000, pp.373-400.

Na, K., B. Jeon, D. Chang, S. Chae, and S. Ann, "Discriminative Training of Hidden Markov Models using Overall Risk Criterion and Reduced Gradient Method," in *Proc. Eurospeech'95*, 1995.

Normandin, Y., "Hidden Markov Models, Maximum Mutual Information Estimation, and the Speech Recognition Proble," *Ph.D Dissertation, McGill University, Montreal,* 1991.

Ortmanns, S., H. Ney and X. Aubert, "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition," *Computer Speech and Language*, 11, 1997, pp.43-72.

Povey, D. and P. C. Woodland, "Minimum Phone Error and I-smoothing for Improved Discriminative Training," in *Proc. ICASSP'02*, 2002.

Povey, D and P. C. Woodland, "Large Scale Discriminative Training of Acoustic Models for Speech Recognition," *Computer Speech and Language*, 16, 2002, pp. 25-47.

Povey, D, "Discriminative Training for Large Vocabulary Speech Recognition," *Ph.D Dissertation, Peterhouse, University of Cambridge*, July 2004.

Povey, D., B. Kingsbury, L. Mangu, G. Saon, H. Soltau and G. Zweig, "FMPE: Discriminatively Trained Features for Speech Recognition," in *Proc. ICASSP'05*, 2005.

Schlüter, R., T. Scharrenbach, V. Steinbiss and H. Ney, "Bayes Risk Minimization using Metric Loss Functions," in *Proc. Eurospeech'05*, 2005.

Schwartz, R. and Y.-L. Chow, "The N-best algorithms: an efficient and exact procedure for finding the N most likely sentence hypotheses," in *Proc. ICASSP'90*, 1990.

Stolcke, A., Y. Konig, M. Weintraub, "Explicit Word Error Minimization in N-best List Rescoring," in *Proc. Eurospeech'97*, 1997.

Stolcke, A., SRI language Modeling Toolkit, version 1.3.3, 2000. *http://www.speech.sri.com/projects/srilm/.*

Wang, H.-M., B. Chen, J.-W. Kuo, and S.-S. Cheng, "MATBN: A Mandarin Chinese Broadcast News Corpus," *International Journal of Computational Linguistics and Chinese Language Processing*, 10(2), 2005, pp.219-236.

Wang, L. and P. C. Woodland, "MPE-Based Discriminative Linear Transform for Speaker Adaptation," in *Proc. ICASSP'04*, 2004.

Zheng, J. and A. Stolcke, "Improved Discriminative Training Using Phone Lattices," in *Proc. INTERSPEECH'05*, 2005.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 223-244

Sense Extraction and Disambiguation for Chinese Words from Bilingual Terminology Bank

Ming-Hong Bai[*,+], Keh-Jiann Chen[*] and Jason S. Chang[+]

Abstract

Using lexical semantic knowledge to solve natural language processing problems has been getting popular in recent years. Because semantic processing relies heavily on lexical semantic knowledge, the construction of lexical semantic databases has become urgent. WordNet is the most famous English semantic knowledge database at present; many researches of word sense disambiguation adopt it as a standard. Because of the success of WordNet, there is a trend to construct WordNet in different languages. In this paper, we propose a methodology for constructing Chinese WordNet by extracting information from a bilingual terminology bank. We developed an algorithm of word-to-word alignment to extract the English-Chinese translation-equivalent word pairs first. Then, the algorithm disambiguates word senses and maps Chinese word senses to WordNet synsets to achieve the goal. In the word-to-word alignment experiment, this alignment algorithm achieves the f-score of 98.4%. In the word sense disambiguation experiment, the extracted senses cover 36.89% of WordNet synsets and the accuracy of the three proposed disambiguation rules achieve the accuracies of 80%, 83% and 87%, respectively.

Keywords: Word Alignment, Word Sense Disambiguation, WordNet, EM Algorithm, Sense Tagging.

1. Introduction

Using lexical semantic knowledge to solve natural language processing problems has been getting popular in recent years. Especially for word sense disambiguation, the semantic lexicon plays a very important role. However, all semantic approaches depend on knowledge of some well established semantic lexical databases which provide semantic information of

[*] Institute of Information Science, Academia Sinica
E-mail: mhbai@sinica.edu.tw; kchen@iis.sinica.edu.tw
[+] Department of Computer Science, National Tsing Hua University
E-mail: jschang@cs.nthu.edu.tw

[Received February 16, 2006; Revised November 28, 2006; Accepted November 30, 2006]

words, such as the different senses of a word, the synonymous or hyperonymy relation between words, etc.

WordNet is a famous semantic lexical database which owns rich lexical information. [Miller 1990]. It not only covers a large set of vocabularies but also establishes a complete taxonomic structure for word senses. Synonymous word senses are grouped into synsets. These synsets are further associated by semantic relations, including hypernyms, hyponyms, holonyms, meronyms, etc. The WordNet has been applied to a wide range of applications, such as word sense disambiguation, information retrieval, computer-assisted language learning, etc. It has apparently become the de facto standard for English word senses now.

Because of the success of WordNet, there is a universally shared interest in construction of WordNet-like and WordNet-embedded lexical databases in different languages. One of the most famous projects is EuroWordNet (EWN). Its goal is to construct a WordNet-like system containing several European languages. Since constructing a WordNet for a new language is a difficult and labor intensive task, using the resources of WordNet to speed up the construction has begun a new trend. Many researchers, such as [Atserias *et al*. 1997], [Daude *et al*. 1999] and [Chang *et al.* 2003], have tried to associate WordNet synsets to other languages automatically with appropriate translations from bilingual dictionaries. The limitation of using bilingual dictionaries as mapping tables for translation equivalences between two languages is the narrow scopes of the dictionaries, since dictionaries usually contain prototypical translations only. For example, the first sense of word "plant" in WordNet is "plant, works, industrial plant"; it was translated as "GongChang"(工廠) in a Chinese-English bilingual dictionary. However, in actual text, it may be also translated as "Chang"(廠), "GongChang"(工場), "ChangFang"(廠房), "suo"(所, such as ‘power plant’/發電所), etc. Various translations, obviously, add complexity and difficulty to map word senses into WordNet synsets.

Instead of using bilingual dictionaries, we adopt a bilingual terminology bank as the semantic lexical database. The latter includes various compound words, in which a word in a different compounding structure may have different translations, thus there are more translation candidates which can be chosen. A bilingual terminology bank has not only helped to avoid the problem of the limited scope of prototypical translations made by common bilingual dictionaries, but has also helped to disambiguate word senses by various translations and collocations [Diab *et al*. 2002], [Bhattacharya 2004]. Nevertheless, using bilingual terminology banks has to face two main challenges: Firstly, we have to deal with the problem of word-to-word alignment for multi-words terms. Secondly, we have to solve the problem of sense ambiguity of the English translation. The approaches for solving these two problems are the major focuses of the paper.

The rest of paper is divided into four sections. Section 2 introduces the resources of this

paper. Section 3 describes the methodology. Experimental setup and results will be addressed in Section 4. A conclusion is provided in Section 5 along with directions for future research.

2. Resources

In this study, we use two dictionaries as the resources to extract semantic information:

a) The Bilingual Terminology Bank from NICT [NICT 2004]

b) A English-Chinese dictionary [Proctor 1988]

The Bilingual Terminology Bank from NICT contains 63 classes of terminologies, with a total of 1,046,058 Chinese terms with their English translations. Among them, 629,352 terms are compounds, which is about 60 percent of the total. The English-Chinese dictionary contains 208,163 words which are used as a supplement. We also adopt WordNet 2.0 as the medium for sense linking. Figure 1 shows some sample entries of the Bilingual Terminology Bank from NICT.

English	**Chinese**	**Class**
succulent stem	肉質莖	Botany
common base current gain	共基電流增益	Electrical Engineering
sliding brush	滑動電刷	Naval Architecture
point of increase	增值點	Mathematics
group carry	成組進位	Computer Science
swine fever	豬瘟	Animal Science
light measurements	光量測	Metrology
reductional grouping	染色體減數分群	Botany
oil film strength	油膜強度	Metrology
normalized quadrature spectrum	標準化四分譜	Meteorology

Figure 1. sample entries of the Bilingual Terminology Bank from NICT.

In English, a compound is usually composed of words and blanks; the latter being a natural boundary to separate words. On the contrary, in Chinese there are no blanks in compound words, so we need to segment words before applying word alignment algorithms. In this paper, we adopt the CKIP Chinese Word Segmentation System, which was developed by the CKIP group of Academia Sinica [CKIP 2006].

3. Methodology

The algorithm can be divided into the following two steps:

1. Find the word to word alignment for each entry in the terminology bank,

2. Assign a synset to the Chinese word sense by resolving the sense ambiguities of its aligned English word.

The first step is to find all possible English translations for each Chinese word, which make it possible to link Chinese words to WordNet synsets. Since the English translation may be ambiguous, the purpose of second step is to employ a word sense disambiguation algorithm to select the appropriate synset for the Chinese word. For example, the term pair (*water tank*, *水 槽*) will be aligned as (*water*/*水 tank*/*槽*) in the first step, so the Chinese word *槽* can be linked to WordNet synsets by its translation *tank*. But *tank* has five senses in WordNet as follows:

*tank*_n_1: an enclosed armored military vehicle,

*tank*_n_2: a large vessel for holding gases or liquids,

*tank*_n_3: as much as a tank will hold,

*tank*_n_4: a freight car that transports liquids or gases in bulk,

*tank*_n_5: a cell for violent prisoners.

The second step is applied to select the best sense translation. In the following subsections, we will describe the detail algorithm of word alignment in section 3.1 and word sense disambiguation in section 3.2.

3.1 Word Alignment

For a Chinese term and its English translation, it is natural to think that the Chinese term is translated from the English term word for word. So, the purpose of word alignment is to connect the words which have a translation relationship between the Chinese term and its English portion. In past years, several statistical-based word alignment methods have been proposed. [Brown *et al*. 1993] proposed a method of word alignment which consists of five translation models, also known as the IBM translation models. Each model focuses on some features of a sentence pair to estimate the translation probability. [Vogel *et al*. 1996] proposed the Hidden-Markov alignment model which makes the alignment probabilities dependent on the alignment position of the previous word rather than on the absolute positions. [Och and Ney 2000] proposed some methods to adjust the IBM models to improve alignment performance.

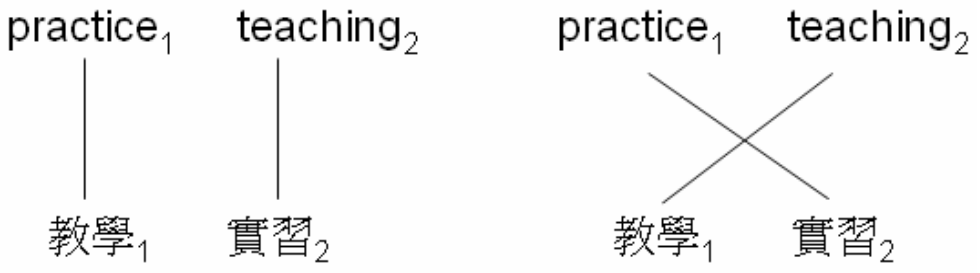
The word alignment task in this paper only focuses on the term pairs of a bilingual terminology bank. Since the length of a term is usually far less than a sentence, some features, such as word position, are no longer important in the task. In this paper, we employ the IBM-1 model, which only focuses on lexical generating probability, to align the words of a bilingual terminology bank.

3.1.1 Modeling Word Alignment

For convenience, we follow the notion of [Brown *et al*. 1993], which defines word alignment as follows:

Suppose we have a English term **e** = $e_1,e_2,\ldots,e_n$ where e_i is an English word, and its corresponding Chinese term **c** = $c_1,c_2,\ldots,c_m$ where c_j is a Chinese word. An alignment from **e** to **c** can be represented by a series **a**=$a_1,a_2,\ldots,a_m$ where each a_j is an integer between 0 and n, such that if c_j is partial (or total) translation of e_i , then $a_j = i$ and if it is not translation of any English word, then a_j=0.

For example, the alignments shown in Figure 2 are two possible alignments from English to Chinese for the term pair (*practice teaching*, *教學 實習*), (a) can be represented by **a**=1,2 while (b) can be represented by **a**=2,1.



(a) (b)

Figure 2. two possible alignments from English to Chinese for the term pair (practice teaching, 教學 實習).

In the word alignment stage, given a pair of terms **c** and **e**, we want to find the most likely alignment **a**=$a_1,a_2,\ldots,a_{\mathrm{m}}$, to maximize the alignment probability P(**a**|**c**,**e**) for the pair. The formula can be represented as follows:

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} P(\mathbf{a} \mid \mathbf{c}, \mathbf{e}), \tag{1}$$

where $\hat{\mathbf{a}}$ is the best alignment of the possible alignments. Suppose we already have lexical translation probabilities for each of the lexical pairs, then, the alignment probability P(**a**|**c**,**e**) can be estimated by means of the lexical translation probabilities as follows:

$$P(\mathbf{a} \mid \mathbf{c}, \mathbf{e}) = \frac{P(\mathbf{a}, \mathbf{c} \mid \mathbf{e})}{P(\mathbf{c} \mid \mathbf{e})} = \prod_{j=1}^{m} P(c_j \mid e_{a_j}) / P(\mathbf{c} \mid \mathbf{e}).$$

The probability of c given e, P(c|e), is a constant for a given term pair (c,e), so formula 1 can be estimated as follows:

$$\hat{\mathbf{a}} = \arg\max_{\mathbf{a}} \prod_{j=1}^{m} P(c_j \mid e_{a_j}) . \quad (2)$$

For example, the probability of the alignment shown in Figure 2 (a) can be estimated by:

$$P(c_1|e_1)P(c_2|e_2)$$
$$= P(\text{教學} \mid practice)\, P(\text{實習} \mid teaching)$$
$$= 0.000480 \text{ x } 1.14\text{x}10^{-13} = 5.48\text{x}10^{-17}.$$

While (b) can be estimated by:

$$P(c_1|e_2)\text{p}(c_2|e_1)$$
$$= P(\text{教學} \mid teaching)P(\text{實習} \mid practice)$$
$$= 0.6953 \text{ x } 0.0940 = 0.0654.$$

In this example, the probability of alignment (b) is larger than (a) in Figure 2. So the alignment (b), (*教學/teaching 實習/practice*), is a better choice than (a), (*教學/practice 實習/teaching*), for the term pair (*practice teaching*, *教學 實習*). The remaining problem of this stage is how to estimate the translation probability $p(c|e)$ for all possible English-Chinese lexical pairs.

3.1.2 Translation Probability Estimation

The method of our translation probability estimation uses the IBM model 1 [Brown *et al*. 1993], which is based on the EM algorithm [Dempster *et al*. 1977], for maximizing the likelihood of generating the Chinese terms, which is the target language, given the English portion, which is the source language. Suppose we have an English term **e** and its Chinese translation **c** in the terminology bank **T**; *e* is a word in **e,** and *c* is a word in **c**. The probability of word *c* given word *e*, $P(c|e)$, can be estimated by iteratively re-estimating the following EM formulae:

Initialization:

$$P(c \mid e) = \frac{1}{|C|}; \quad (3)$$

E-step:

$$Z(c, e; \mathbf{c}, \mathbf{e}) = \sum_{\forall \mathbf{a}} P(\mathbf{a} \mid \mathbf{c}, \mathbf{e}) \sum_{j=1}^{m} \delta(c, c_j)\delta(e, e_{a_j}) , \quad (4)$$

$$P(\mathbf{a} \mid \mathbf{c}, \mathbf{e}) = \frac{P(\mathbf{a}, \mathbf{c} \mid \mathbf{e})}{\sum_{\forall \mathbf{a}'} P(\mathbf{a}', \mathbf{c} \mid \mathbf{e})} = \frac{\prod_{j=1}^{m} P(c_j \mid e_{a_j})}{\sum_{\forall \mathbf{a}'} \prod_{j=1}^{m} P(c_j \mid e_{a'_j})}; \quad (5)$$

M-step:

$$P(c \mid e) = \frac{\sum_{t=1}^{|T|} Z(c, e; \mathbf{c}^{(t)}, \mathbf{e}^{(t)})}{\sum_{\forall v \in C} \sum_{t=1}^{|T|} Z(v, e; \mathbf{c}^{(t)}, \mathbf{e}^{(t)})}. \tag{6}$$

In the EM training process, we initially assume that the translation probability for any Chinese word *c* given English word *e*, *P*(*c*|*e*), is uniformly distributed as in formula 3, where ***C*** denotes the set of all Chinese words in the terminology bank. In the E-step, we estimate the expected number of times that *e* connects to *c* in the term pair (**c**,**e**). As in formula 4, we sum up the expected counts of the connection from *e* to *c* over all possible alignments which contain the connection. Formula 5 is the detailed definition of the probability of an alignment **a** given (**c**,**e**). Usually, it is hard to evaluate the formulae in E-step. Fortunately, it has been proven [Brown *et al*. 1993] that the expectation formulae, 4 and 5, can be merged and simplified as follows:

$$\begin{aligned} Z(c, e; \mathbf{c}, \mathbf{e}) &= \sum_{\mathbf{a}} P(\mathbf{a} \mid \mathbf{c}, \mathbf{e}) \sum_{j=1}^{m} \delta(c, c_j) \delta(e, e_{a_j}) \\ &= \frac{\sum_{\mathbf{a}} \prod_{j=1}^{m} P(c_j \mid e_{a_j}) \sum_{j=1}^{m} \delta(c, c_j) \delta(e, e_{a_j})}{\sum_{\forall \mathbf{a}'} \prod_{j=1}^{m} P(c_j \mid e_{a'_j})} \\ &= \frac{P(c \mid e) \prod_{j=1, c_j \neq c}^{m} \sum_{i=0, e_i \neq e}^{n} P(c_j \mid e_i)}{\prod_{j=1}^{m} \sum_{i=0}^{n} P(c_j \mid e_i)} \sum_{j=1}^{m} \delta(c, c_j) \sum_{i=0}^{n} \delta(e, e_i) \\ &= \frac{P(c \mid e)}{\sum_{i=0}^{n} P(c \mid e_i)} \sum_{j=1}^{m} \delta(c, c_j) \sum_{i=0}^{n} \delta(e, e_i). \end{aligned} \tag{7}$$

After merging and simplifying, as formula 7, the E-step becomes very simple and effective for computing.

In the M-step, we re-estimate the translation probability, *P*(*c*|*e*). As shown in formula 6, we sum up the expected number of connections from *e* to *c* over the whole bank divide by the expected number of *c*.

The training process will count the expected number, E-step, and re-estimate the translation probability, M-step, iteratively until it has converged.

For instance, as the example shown in Figure 2, the English term **e**= *practice teaching* and Chinese term **c**=*教學 實習* are given. Assume the total number of Chinese words in the terminology bank is 100,000. Initially, the probabilities of each translation are as follows:

$$P(\ 教學\ |practice) = \frac{1}{|C|} = 0.00001, \qquad P(\ 教學\ |teaching) = \frac{1}{|C|} = 0.00001,$$

$$P(\ 實習\ |practice) = \frac{1}{|C|} = 0.00001, \qquad P(\ 實習\ |teaching) = \frac{1}{|C|} = 0.00001.$$

In E-step, we count the expected number for all possible connections in the term pair:

$$Z(\ 教學\ , practice; \mathbf{e}, \mathbf{c}) = \frac{P(教學 | practice)}{P(教學 | practice) + P(教學 | teaching)} = 0.5,$$

$$Z(\ 教學\ , teaching; \mathbf{e}, \mathbf{c}) = \frac{P(教學 | teaching)}{P(教學 | practice) + P(教學 | teaching)} = 0.5,$$

$$Z(\ 實習\ , practice; \mathbf{e}, \mathbf{c}) = \frac{P(實習 | practice)}{P(實習 | practice) + P(實習 | teaching)} = 0.5,$$

$$Z(\ 實習\ , teaching; \mathbf{e}, \mathbf{c}) = \frac{P(實習 | practice)}{P(實習 | practice) + P(實習 | teaching)} = 0.5.$$

In M-step, we first count the global expected number of each translation by summing up the expected number of each data entry over the whole term bank:

$$\sum_{t=1}^{|T|} Z(教學, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)}) = 0.7,$$

$$\sum_{t=1}^{|T|} Z(教學, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)}) = 43.72,$$

$$\sum_{t=1}^{|T|} Z(實習, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)}) = 5.37,$$

$$\sum_{t=1}^{|T|} Z(實習, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)}) = 0.95.$$

After the global expected number of each translation has been counted, we can re-estimate the translation probabilities by means of the expected numbers:

$$P(\ 教學\ |practice) = \frac{\sum_{t=1}^{|T|} Z(教學, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})}{\sum_{v \in C} \sum_{t=1}^{|T|} Z(v, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})} = \frac{0.7}{110.67} = 0.00632,$$

$$P(\ 教學\ |teaching) = \frac{\sum_{t=1}^{|T|} Z(教學, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})}{\sum_{v \in C} \sum_{t=1}^{|T|} Z(v, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})} = \frac{43.72}{121.88} = 0.35871,$$

$$P(\textit{實習} \mid practice) = \frac{\sum_{t=1}^{|T|} Z(\textit{實習}, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})}{\sum_{v \in C} \sum_{t=1}^{|T|} Z(v, practice; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})} = \frac{5.37}{110.67} = 0.04852,$$

$$P(\textit{實習} \mid teaching) = \frac{\sum_{t=1}^{|T|} Z(\textit{實習}, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})}{\sum_{v \in C} \sum_{t=1}^{|T|} Z(v, teaching; \mathbf{e}^{(t)}, \mathbf{c}^{(t)})} = \frac{0.95}{121.88} = 0.00779.$$

The training process will count the expected number and re-estimate the translation iteratively until it has converged. There are some translation probabilities estimated in this experiment shown in Figures 3-6.

English	Chinese	P(*c* | *e*)
water	水	0.599932
water	水位	0.048781
water	水分	0.011677
water	用水	0.011427
water	地下水	0.010800
water	水壓	0.009310
water	水量	0.007905
water	水管	0.007640
water	位	0.007471
water	水面	0.006704

Figure 3. translation probabilities for water.

English	Chinese	P(*c* | *e*)
tank	槽	0.292606
tank	櫃	0.176049
tank	艙	0.077515
tank	箱	0.034325
tank	水	0.025067
tank	液	0.018411
tank	水槽	0.016570
tank	池	0.016157
tank	罐	0.015687
tank	水箱	0.012206

Figure 4. translation probabilities for tank.

English	Chinese	P(*c* | *e*)
practice	練習	0.163636
practice	實習	0.093320
practice	演習	0.058102
practice	實務	0.056980
practice	操作	0.051331
practice	優良	0.042036
practice	作業	0.038144
practice	方法	0.036161
practice	實作	0.034805
practice	實際	0.025800

Figure 5. translation probabilities for practice.

English	Chinese	P(*c* | *e*)
teaching	教學	0.698757
teaching	教學法	0.137614
teaching	教材	0.045780
teaching	單元	0.015502
teaching	教具	0.010315
teaching	教導	0.007246
teaching	教會	0.007246
teaching	教授	0.007246
teaching	教訓	0.007246
teaching	教	0.007246

Figure 6. translation probabilities for teaching.

3.1.3 Imposing Alignment Constraints

As was mentioned in Section 3.1.1, the goal of word alignment is to find the best alignment candidate to maximize the translation probability of a term pair. However, in real situations there are some problems that have to be solved:

1. Cross connections: assume there is a series of words, c_j,c_{j+1},c_{j+2} in a Chinese term, if c_j and c_{j+2} connect to the same English word while c_{j+1} connects to any other word, we call this

alignment contains a cross connection. There is an example of cross connection shown in Figure 7. The Chinese word *校* is more likely to connect to *examination* shown in Figure 8.

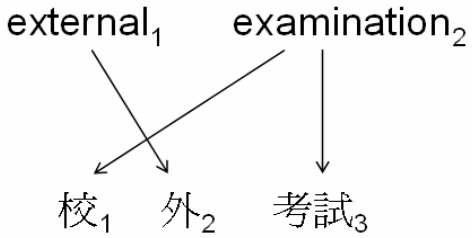


Figure 7. example of cross connection, 校 and 考試 connected to examination while 外 connected to external.

	校	外	考試
external	$1.4x10^{-7}$	**0.575537**	$5.3x10^{-9}$
examination	**$5.2x10^{-6}$**	5.2x10-6	**0.172751**

Figure 8. example of cross connection: the translation probabilities of the example, it shows that 校 is more likely to connect to examination.

2. Function words: in word alignment stage, function words are usually ignored except when they are part of compound words. For example, Figure 9, *of* is a part of a compound which can not be skipped, while in Figure 10, *of* can be skipped.

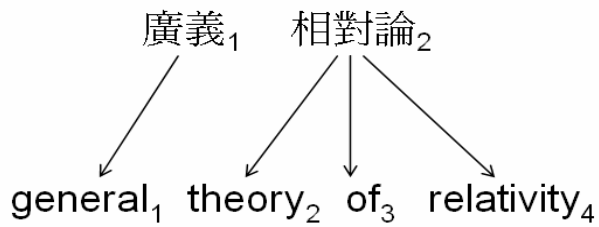


Figure 9. of is part of compound.

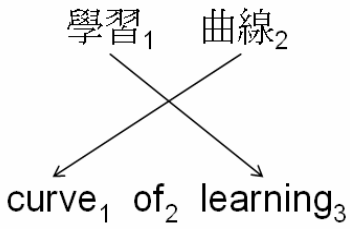


Figure 10. of is not part of compound.

In order to solve this problem, two constraints are imposed on the alignment algorithm. Formula 1 is altered by using a cost function instead of probability, defined as follows:

$$\mathbf{a} = \arg\min_{\mathbf{a}} cost(\mathbf{a}), \tag{8}$$

where cost function is given by:

$$\cos t(\mathbf{a}) = \begin{cases} \infty, & \text{if } cross\_connection(\mathbf{a}) = true \\ \infty, & \text{if } a_i \text{ connects } c_i \text{ to any word and } c_i \text{ is a function word and } c_i \text{ is not part of compound} \\ \sum_{j=1}^{k} -\log(p(c_j \mid e_{a_j})) & else \end{cases} . \quad (9)$$

The *cross connection* function is used to detect the cross connection in an alignment candidate. If a cross connection is found, the alignment candidate will be assigned a large cost value. The function was given by:

$$cross\_connection(\mathbf{a}) = \begin{cases} true, & \text{if } a_i \neq a_{i+1} \text{ and } a_i = a_{i+2} \\ false, & else \end{cases} . \quad (10)$$

3.1.4 Connection Directions

There are two connection directions in word alignment: from Chinese to English, (where Chinese is the source language while English is the target language), and from English to Chinese. The alignment method of the IBM models has a restriction; a word of target language can only be connected to exactly one word of the source language. This restriction causes two words in the source language not to be able to connect to a word in the target language.

For example, in Figure 11, for alignment from Chinese to English, *cedar* should be connected to both 雪 and 松, but the model does not allow the connection in this direction. Figure 12 is another example of the same problem from English to Chinese.

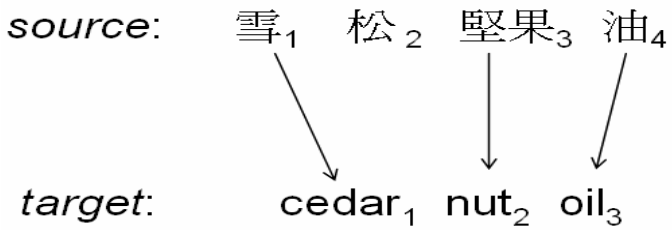


Figure 11. cedar can not be connected by both 雪 and 松 in this direction.

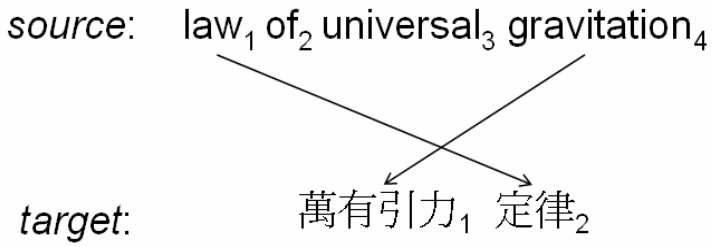


Figure 12.萬有引力 can not be connected by both universal and gravitation in this direction.

In order to solve this problem, the alignments of these two directions are merged using the following steps: 1. Align from Chinese to English. Each word of an English compound will be connected by the same Chinese word in this step which will be treated as an alignment unit in the next step. 2. Align from English to Chinese. Each word of a Chinese compound will be connected to the same English unit, a word or merged compound, in this step.

For example, *universal gravitation* was merged in step 1 while *雪* and *松* were not merged in the same step, as shown in Figure 13. In step2, *雪* and *松* were merged and *universal gravitation* will be treated as a unit in the same step, as shown in Figure 14.

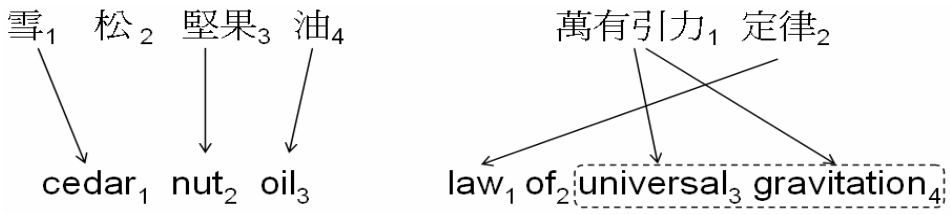


Figure 13. 雪 and 松 were not merged in step 1 while universal gravitation was merged in the same step.

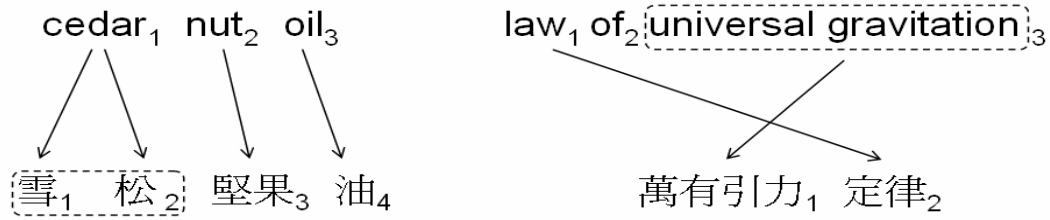


Figure 14. step 2, 雪 and 松 were merged in step 2 and universal gravitation was treated as a unit in the same step.

After these two steps, all of the compounds in each language will be merged. Figure 15 shows some examples of word alignment in these experiments.

English Term	Chinese Term	Alignment
evaporation tank	蒸發 槽	evaporation/蒸發 tank/槽
wind-wave tank	風浪 水槽	wind-wave/風浪 tank/水槽
wave tank	波浪 水槽	wave/波浪 tank/水槽
volumetric tank	量 水箱	volumetric/量 tank/水箱
curve of learning	學習 曲線	curve/曲線 of/ learning/學習
exchange of students	學生 交換	exchange/交換 of/ students/學生
practice teaching	教學 實習	practice/實習 teaching/教學
wall cloud	雲 牆	wall/牆 cloud/雲
gas mixture	混合 氣體	gas/氣體 mixture/混合
air choke valve	阻 氣 閥	air/氣 choke/阻 valve/閥

Figure 15. some examples of word alignment.

3.2 Sense Tagging

When we tag Chinese words with WordNet senses, if the translation of a word has only one sense, a monosemous word, it can be tagged with that sense directly. If the translation has more than one sense, we should use a disambiguation method to get the appropriate sense. In past years, a lot of word sense disambiguation (WSD) methods have been proposed, including supervised, bootstrapping, and unsupervised. Supervised and bootstrapping methods usually resolve an ambiguity in the collocations of the target word, which implies that the target word should be in a complete sentence. These are not appropriate for this project's data. When some statistical based unsupervised methods are not accurate enough, they will add too much noise to the results. For the purpose of building a high quality dictionary, we tend to use a high precision WSD method which should also be appropriate for a bilingual term bank. We employ some heuristic rules, which are motivated by [Atserias *et al*. 1997], described as follows:

Heuristic 1.

If e_i is a morpheme of **e** then pick the sense of e_i, say s_j, which contains hyponym **e**.

This heuristic rule works for head morphemes of compounds. For example, as shown in figure 16, the term pair (*water tank*, *水 槽*) is aligned as (*water*/*水 tank*/*槽*). There are five senses for *tank*. The above heuristic rule will select *tank-2* as the sense of *tank*/*槽* because there is only one sense of *water tank* and the sense is a hyponym of *tank-2*. In this case, the sense of *water tank* can be tagged as *water tank-1* and *tank* can be tagged as *tank-2*.

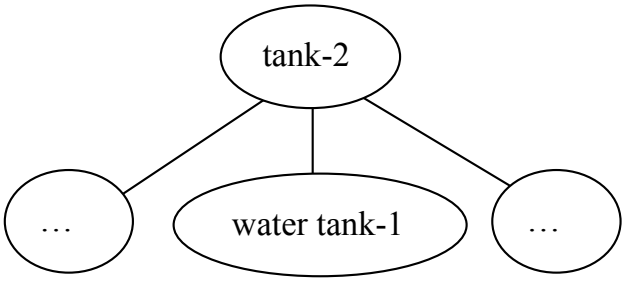


Figure 16. water tank-1 is a hyponym of tank-2.

Heuristic 2.

Suppose the set $\{e_1,e_2,\ldots,e_k\}$ contains all possible translations of Chinese word *c*,

Case 1: If $\{e_1,e_2,\ldots,e_k\}$ share a common sense s_t, then pick s_t as their sense.

Case 2: If one element of the set $\{e_1,e_2,\ldots,e_k\}$, say e_i, has a sense s_t which is the hypernym of synsets corresponding to the rest of the words. We say that they nearly share the same sense and pick s_t as the sense e_i, pick the corresponding hyponyms as the sense of the rest of words.

An example of case 1 is the translations of *腳踏車*, {*bicycle*, *bike*, *wheel*}, which are a subset of a synset. This means that the synset is the common sense of these words and we can pick it as the words' sense. An example of case 2, as shown in figure 17, is the translations of *信號旗*, {*signal*, *signal flag*, *code flag*}, although these words do not exactly share the same sense, one sense of *signal* is the hypernym of *signal flag* and *code flag*. This means that they nearly share the same sense; we pick the hypernym, *signal-1*, as the sense of *signal* and the corresponding hyponyms as the sense of *signal flag* and *code flag*.

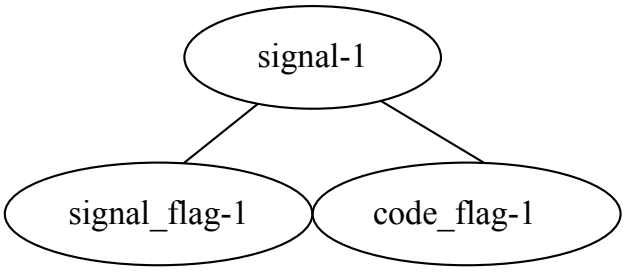


Figure 17. the translations of 信號旗, {signal, signal flag, code flag}, are nearly share the same sense.

Heuristic 3.

If some of the translations of *c* are tagged in the previous steps and the results show that the translations of *c* is always tagged with the same sense, we think *c* to have mono sense, so pick that sense as the sense of untagged translations.

In the previous steps, many Chinese-English pairs have been tagged with WordNet senses. In these tagged instances, we found that some Chinese words were always tagged with the same synset, although they may have many different English translations, and these English words may be ambiguous themselves. The untagged translations of the Chinese word can be tagged with the same synset.

For example, as shown in Figure 18, *防波堤* has many different translations and some of them are ambiguous in WordNet, (*groin* has 3 senses in WordNet). In fact, those seemingly different senses tagged by previous steps actually are indexed by the same synset in WordNet, so we guess that *防波堤* has mono sense and will be tagged the same synset for all instances.

Chinese word	English word	Sense
防波堤	breakwater	breakwater-1
防波堤	groin	groin-2
防波堤	groyne	groyne-1
防波堤	mole	mole-5
防波堤	bulwark	bulwark-3
防波堤	seawall	seawall-1
防波堤	jetty	jetty-1

Figure 18. the possible translations of 防波堤 and its sense tagged by the previous steps.

4. Experiments

In the experiment of word alignment, we extract 840,187 English-Chinese translation pairs which contain 445,830 Chinese word types and 318,048 English word types. On average, each Chinese word has 1.88 English translations while each English word has 2.64 Chinese translations.

In word sense disambiguation, 124,752 Chinese words were linked to 42,589 WordNet synsets, which contain 165,775 (Chinese word, synset) translation pairs. On average, each Chinese word was discovered to have 1.33 senses in terms of WordNet synsets. In the following subsection, we will evaluate the performance of the word alignments and WSD results.

4.1 Results of Word Alignment

In order to evaluate the performance of word alignment, we randomly select 500 term pairs from a terminology bank and align them manually as the gold standard, As single-morpheme terms do not need to be aligned, compound words were considered only. We follow the

evaluation method defined by [Och and Ney 2000], which defined precision, recall and alignment error rate (AER) as follows:

$$\text{recall} = \frac{|A \cap S|}{|S|},$$

$$\text{precision} = \frac{|A \cap P|}{|A|},$$

$$\text{AER} = 1 - \frac{|A \cap S| + |A \cap P|}{|A| + |S|},$$

where S denotes the annotated set of sure alignments, P denotes the annotated set of possible alignments, and A denotes the set of alignments produced by the alignment method.

The results are shown in Table 1. The recall and precision figures show that the word alignment results are quite accurate. As we expected, the word alignment in phrases is much easier and accurate than in complete sentences. Note that the f-scores of word alignment tasks in complete sentences, even the current state-of-the-art alignments for naturally related languages such as English and French, are still less than 95 [Blunsom *et al*. 2006].

Table 1. the performance of our word alignment method.

recall	precision	f-score	AER
98.2	98.6	98.4	1.6

Table 2. typical errors of word alignment.

Error Type	Error Samples
Word Segmentation	half-wave/半 length/波長 criterion/準則 spiral/螺旋 coal/煤機 cleaner/洗 american/西 ginseng/洋參 second/再 wind/生氣 microlen/微透鏡藕 coupler/合器 atomic/原子能 energy/階
transliteration	san/聖胡 julian/連安
asymmetric translation	navigation/航行參考 star/星
abbreviation	double/ III/托克馬克熱核反應器

The main alignment errors are caused by the following reasons as shown in Table 2. The first error type was caused by the errors of word segmentation. For example, *西洋參* should be segmented as *西洋 參* instead of *西 洋參* and *再生氣* should be segmented as *再生氣* instead of *再 生氣*. The second error type was the mapping of transliterations which is a different type of word alignment. The third type was caused by the asymmetric translation of

the data. For example, in the term pair (navigation star, *航行 參考 星*), the Chinese word *參考* has no appropriate mapping in the English portion. The fourth type was caused by abbreviation which is also a difficult problem in regards to word alignment.

4.2 Result of Word Sense Disambiguation

Since the goal of these experiments is to build a Chinese WordNet automatically, we concerned more with the quality of WSD than the quantity. To evaluate the accuracy of these heuristic rules, we randomly selected 200 sense tagged words for each heuristic rule and checked the sense of each word manually. The accuracy rate of WSD results are defined as follows:

$$\text{accuracy rate} = \frac{\text{\# of selected words with correct sense}}{\text{\# of selected words}}.$$

The accuracy of each heuristic rule is shown in Table 3. It shows that the accuracy of heuristic rules is all over 80 %. Note that, in the lexical sample tasks of Senseval 3 [Mihalcea *et al*. 2004], the precision of the best supervised WSD methods is less than 73%, the unsupervised methods are even worse. Furthermore, these methods depend highly on the contexts of target words, which is not suitable in these experiments. These are the reasons why we use the heuristic rules instead of conventional WSD methods.

Table 3. Disambiguation accuracy of each heuristic rule.

	# words	#words with correct sense	accuracy rate
Heuristic 1	200	160	80.0 %
Heuristic 2	200	167	83.5 %
Heuristic 3	200	174	87.0 %

We also concerned with how many WordNet senses can be linked with Chinese words. There are two coverage rates, defined as follows:

$$\text{coverage rate of word-sense pairs} = \frac{\text{\# of word sense pairs are linked}}{\text{\# of word sense pairs in WordNet}},$$

$$\text{coverage rate of synsets} = \frac{\text{\# of synsets are linked}}{\text{\# of synsets in WordNet}}.$$

In the WSD steps, 484,771 tokens are tagged with WordNet synsets, in which 54,654 distinct word-sense pairs are contained. In other words, there are 54,654 distinct word-sense pairs which are linked with any Chinese word. The coverage of word-sense pairs and synsets are shown in Table 4. The synset coverage of heuristic rule 3 is not listed in the table, because it just tags the Chinese words which have been disambiguated in the previous steps and does

not link any Chinese word with new synset. The table shows that the coverage of word-sense pairs in WordNet 2.0 is 26.9% and the coverage of synsets is 36.89 %.

Table 4. the coverage of each heuristic rule in WordNet 2.0.

	#tokens	#word-sense pairs	word-sense pair coverage	#synsets	synset coverage
monosemous word	370,991	48,623	23.94 %	39,953	34.61 %
Heuristic 1	29,422	4,211	2.07 %	3,452	2.99 %
Heuristic 2	29,311	2,050	1.00 %	1,685	1.46 %
Heuristic 3	81,734	1,931	0.95 %	-	-
Total	484,771	54,654	26.90 %	42,589	36.89 %

It seems the coverage of the experiments is too low. One possible reason is that most of the synsets in WordNet are infrequent. To prove this phenomenon, we use the frequencies of each sense provided by WordNet, which are the occurrence frequencies for each synset in the SemCor Corpus. As per analysis, there are 115,423 synsets in WordNet 2.0, but only 28,688 (24.8%) synsets appear in the SemCor. It shows that most of the senses are low frequency senses in WordNet.

Another issue is that, the coverage is contributed mostly by monosemous words. About 17% of words are ambiguous in WordNet. It seems that there is still room to improve.

5. Conclusions and Future Researches

In this paper, we propose a methodology to extract Chinese-English translation pairs from a large-scale bilingual terminology bank, and link the translation pairs to WordNet synsets. We faced two problems in this study: 1. Word-to-word alignment for each entry in the terminology bank, which helps to extract corresponding English translations for each Chinese word. 2. Word sense disambiguation, which helps to select the appropriate sense when the English translation of a Chinese word is ambiguous.

The evaluation of the experiments shows that the f-score of word alignment archives 98.4%. In the word sense disambiguation stage, the word-sense pairs extracted from the terminology bank cover 26.9% of WordNet word-sense pairs. Also, the distinct senses cover 36.89% of WordNet synsets. The accuracy of the three heuristic rules achieves 80%, 83 %, and 87 %.

A bilingual terminology bank provides some advantages over a bilingual parallel corpus for extracting information. For example, we can extract more Chinese-English translation pairs through the various appearances of a word which is contained in different compounds. The other advantage is that most of compound words in terminology bank are composed of

only 2-3 words, which results in the word alignment accuracy of a terminology bank being much higher than a bilingual corpus.

In the future we will try to use some other word sense disambiguation methods to increase the coverage of words and senses in WordNet and to extract more information from terminology bank.

References

Atserias, J., S. Climent, X. Farreres, G. Rigau and H. Rodríguez, "Combining Multiple Methods for the Automatic Construction of Multilingual WordNets," In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*, 1997, Tzigov Chark, Bulgaria, pp. 143-149.

Bhattacharya, I., L. Getoor and Y. Bengio, "Unsupervised Sense Disambiguation Using Bilingual Probabilistic Models, " In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*, 2004, Barcelona, Spain, pp. 287-294.

Blunsom, P. and T. Cohn, "Discriminative Word Alignment with Conditional Random Fields," In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, 2006, Sydney, Australia, pp. 65-72.

Brown, P.F., S.A.D. Pietra, V.J.D. Pietra, and R.L. Mercer, "The Mathematics of Machine Translation: Parameter Estimation," *Computational Linguistics*, 19(2), 1993, pp. 263–311.

Chang, J.S., T. Lin, G.-N. You, T.C. Chuang and C.-T. Hsieh, "Building A Chinese WordNet Via Class-Based Translation Model," *International Journal of Computational Linguistics and Chinese Language Processing*, 8(2), 2003, pp. 61-76.

CKIP, Chinese Word Segmentation System, http://ckipsvr.iis.sinica.edu.tw/, 2006

Daudé, J., L. Padró and G. Rigau, "Mapping Multilingual Hierarchies using Relaxation Labelling," In *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999, College Park, Maryland.

Dempster, A.P., N.M. Laird and D.B. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 39(1), 1977, pp. 1-38.

Diab, M. and P. Resnik, "An Unsupervised Method for Word Sense Tagging using Parallel Corpora," In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 2002, NJ, USA, pp. 255-262.

Mihalcea, R. and T. Chklovski, "The SENSEVAL–3 English Lexical Sample Task," In *Proceedings of Senseval-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, Barcelona, Spain, pp. 25-28.

Miller, G., "WordNet: An online lexical database," *International Journal of Lexicography*, 3(4), 1990, pp. 235-312.

NICT, 學術名詞資訊網, http://terms.nict.gov.tw/, 2006.

Och, F.J. and Hermann N., "Improved Statistical Alignment Models," In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, 2000, Hong Kong, pp. 440-447.

Proctor, P., "Longman English-Chinese Dictionary of Contemporary English," *Longman Group (Far East) Ltd*., Hong Kong, 1988.

Vogel, S., H. Ney, C. Tillmann, "HMM-based word alignment in statistical translation," In *Proceedings of the 16th conference on Computational linguistics*, 1996, Morristown, NJ, pp. 836-841.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 245-280

A Probe into Ambiguities of Determinative-Measure Compounds

Shih-Min Li*,+, Su-Chu Lin*, Chia-Hung Tai*, and Keh-Jiann Chen*

Abstract

This paper aims to further probe into the problems of ambiguities for automatic identification of determinative-measure compounds (DMs) in Chinese and to develop sets of rules to identify DMs and their parts of speech. It is known that Chinese DMs are identifiable by regular expressions. DM rule matching helps one solve word segmentation ambiguities, and parts of speech help one improve sense recognition and part-of-speech tagging. In this paper, a deep analysis based on corpus data was studied. With analyses of error identification and disambiguation of DM compounds, the authors classified three types of ambiguities, *i.e.* word segmentation, sense, and pos ambiguities. DM rules are necessary complements to dictionaries and helpful to resolve word segmentation ambiguities by applying resolution principles and segmentation models. Sense and pos ambiguities are also expected to be resolved by different approaches during postprocessing.

Keywords: Ambiguities, Word Segmentation Ambiguities, Sense Ambiguities, Part-of Speech Ambiguities, Determinative-Measure Compounds

1. Introduction

To a speaker of English, one of the most striking features of the Mandarin noun phrase is the classifier. A classifier is a word that must occur with a number and/or a demonstrative, or certain quantifiers before the noun [Li and Thompson 1981]. Furthermore, Li and Thompson [1981] assert that any measure word can be a classifier, so the combination of demonstrative and/or number or quantifier plus a classifier or a measure is defined as a classifier phrase or a measure phrase. For example, *san ge* in *san ge ren* 'three people' (三個人), *zhe zhan* in *zhe zhan deng* 'this lamp' (這盞燈), *ji jian* in *ji jian yifu* 'a few / how many garments' (幾件衣服), *liu li* in *liu li lu* 'six miles of road' (六里路), *na jin* in *na jin yangrou* 'that tael of lamb'

* Institute of Information Science, Academia Sinica, Taipei
E-mail: {shihmin, jess}@hp.iis.sinica.edu.tw; {glaxy, kchen}@iis.sinica.edu.tw
+ Graduate Institute of Linguistics, National Chengchi University, Taipei
E-mail: 95555501@nccu.edu.tw

[Received February 16, 2006; Revised November 27, 2006; Accepted November 28, 2006]

(那斤羊肉) and *ji gang* in *ji gang cu* 'a few / how many vats of vinegar' (幾缸醋) are classifier phrases / measure phrases, which are regarded as Determinative-Measure (DM) compounds in Chao [1968]. A determinative (D) and a measure normally make a compound with unlimited versatility and form a transient word of no lexical import [Chao 1968]. Although the demonstratives, numerals, and measures may be listed exhaustedly, their combination is inexhaustible. It is impossible to list thoroughly all combinations of DMs in dictionaries. Therefore, it requires a representational model to express DM compounds in Chinese NLP.

In Chinese, word segmentation, sense, and pos (*i.e.* part-of-speech) ambiguities commonly occur in certain constructions of DMs or DM-like structures. For examples:

(1) *三個*月餅舖的銷售量

a. *sange yuebingpu de xiaoshouliang*
three-M moon-cake store DE sales volume
three moon-cake stores' sales volume

b. *sangeyue bingpu de xiaoshouliang*
three-M months cake store DE sales volume
the cake store's three-month sales volume

(2) 取*此名*

a. *qu ciming*
choose this-M
choose this one (person)

b. *qu ci ming*
name this name
name it this name

(3) *二十五年*的審核、排隊、等待

a. *ershiwunian de shenhe paidui dengdai*
twenty five-M DE examine line up wait
examining, lining up and waiting in the year of twenty five

b. *ershiwunian de shenhe paidui dengdai*
twenty five-M DE examine line up wait
examining, lining up and waiting for twenty five years

The DM compound *sange* in example (1) modifies moon-cake stores as well as months. The string *sangeyuebingpu* can be segmented into either *sange yuebingpu* or *sangeyue bingpu*, which has word segmentation ambiguity. In example (2), *ming* functions either as a measure or as a noun. Although example (2) has two meanings and is sense ambiguous, the roles assigned to *ciming* in (2a) and (2b) are both the goal. In example (3), *ershiwunian* is a time referent, and it can either be a time point specifying the event-time of the verb or denote the period of time delimitating the time length of the event. In Sinica Treebank, no matter whether *ershiwunian* behaves as a time point or a time length, *ershiwunian* is tagged as a unit. When *ershiwunian* behaves as a time point, its pos is Ndaad and semantic role is time[1]. Furthermore, when *ershiwunian* behaves as a time length, its pos is DM and semantic role is duration. However, according to CKIP's word segmentation standard of Sinica Corpus, the temporal and locative DM structures are combined together when the meaning of the structure is not obtained from the composition of the components of the structure. Therefore, the temporal DM *ershiwunian* in (3a) is combined as a unit and tagged as Nd in Sinica Corpus while that in (3b) is segmented into two units, *i.e*. *ershiwu* and *nian*, and tagged as Neu and Nf individually. Therefore, example (3) has pos ambiguity. The different degrees of ambiguities are shown in examples (1) to (3).

In this paper, we examine and analyze Mandarin Chinese DMs in Sinica Corpus and Sinica Treebank. In section 3, we introduce the regular expression approach to identify DMs and their poses. In section 4, we make a study of the structures and ambiguities of DMs, and then try to analyze and disambiguate these DMs. Section 5 is for implementation and evaluation.

2. Literature Review

To deal with DMs, first one must give a proper definition to DMs. Thus, one can delimit the scope of the discussion. There are numerous discussions on determinatives as well as measures, especially on the types of measures.[2] The classification of measures is beyond the scope of this paper. To avoid confusion between classifiers and measures, one must pay attention to the relation between them. Tai [1994] asserts that in the literature on general grammar as well as Chinese grammar, classifiers and measures words are often treated together under one single framework of analysis. Chao [1968] treats classifiers as one kind of measure. In his definition, a measure is a bound morpheme which forms a DM compound with

[1] All the symbols such as Ndaad will be defined in the appendix of this paper.

[2] Chao [1968] and Li and Thompson [1981] detect measures and classifiers. He [2000] traces the diachronic names of measures and mentions related literature on measures. The dictionary of measures pressed by Mandarin Daily News Association and CKIP [1997] lists all the possible measures in Mandarin Chinese.

one of the determinatives enumerated above [Chao 1968]. Classifiers are defined as 'individual measures', which is one of the nine kinds of measures. As was mentioned in the section of introduction, Chao considers that determinatives are listable and measures are largely listable so D and M can be defined by enumeration, and that DM compounds have unlimited versatility. However, Li and Thompson [1981] blend classifiers with measures. They conclude that, not only does a measure word generally not take a classifier, but also any measure word can be a classifier. In Tai's opinion [1944], in order to better understand the nature of categorization in a classifier system, it is not only desirable but also necessary to differentiate classifiers from measure words. These studies on the distinction between classifiers and measures are not very clear-cut. In this paper, we discuss ambiguities of DMs in NLP as well as adopt the CKIP DM rules and symbols of poses, and therefore inherit the definition of determinative-measure compounds (DMs) in Mo *et al.* [1991]. Mo *et al*. define a DM as the composition of one or more determinatives together with an optional measure. The definition of Mo *et al.* is used to apply to NLP and somewhat different from traditional linguistics definition.

As for ambiguity, Crystal [1991] specifies that the general sense of ambiguity is a word or sentence which expresses more than one meaning. The most widely discussed type of ambiguity in recent years has been grammatical (or structural) ambiguity. In the structure *new houses and shops*, it could be analysed either as *new* [*houses and shops*] (*i.e*. both are new) or [*new houses*] *and shops* (*i.e*. only the houses are new). Furthermore, according to Crystal's assertion, ambiguity which does not arise from the grammatical analysis of a sentence, but is due solely to the alternative meanings of an individual lexical item, is referred to as lexical ambiguity, *e.g. I found the table fascinating* (= 'object of furniture' or 'table of figures'). Moreover, the definition of structural and lexical ambiguities can be referred to Prins [2005]. Prins mentions if one restricts his or her attention to the syntax in texts, then one may focus on ambiguity in two forms. The first is lexical ambiguity, the second is structural ambiguity. Lexical ambiguity arises when one word can have several meanings. Structural ambiguity arises when parts of a sentence can be syntactically combined in more than one way. Prins believes humans can resolve most ambiguity, of both types, without even being consciously aware of the alternatives. The remaining ambiguity, of which we are aware, is resolved when knowledge about the world is used in combination with what is known about the linguistic context of the ambiguity to arrive at the most likely analysis. Jurafsky and Martin [2000] define ambiguity as a sentence or words which can have more than one parse. Deciding which category a word belongs to can be solved by part-of-speech tagging. Deciding what sense a word has can be solved by word sense disambiguation. Resolutions of part-of-speech and word sense ambiguities are two important kinds of lexical disambiguation [Jurafsky and Martin 2000]. Furthermore, structural ambiguity occurs when the grammar assigns more than

one possible parse to a sentence. Structural disambiguation / syntactic disambiguation can be addressed by probabilistic parsing. Three particularly common kinds of structural ambiguity are attachment ambiguity, coordination ambiguity, and noun-phrase bracketing ambiguity [Jurafsky and Martin 2000]. In the following analysis, we find that Prins' division of ambiguity into structural ambiguity and lexical ambiguity is not enough to deal with ambiguities in NLP. One must apply Jurafsky and Martin's classification to obtain more detailed discussion on ambiguities of DMs in NLP. Word segmentation ambiguity caused by different segmentation of words is a kind of structural ambiguity. With the same word segmentation, that string of words may still be ambiguous because the string may either have more than one meaning or have different parts of speech, semantic roles and functions. Therefore, weone have sense ambiguity and pos ambiguity, which are the two subtypes of lexical ambiguities of DMs.

3. Regular Expression Approach for Identifying DMs

In this section, we introduce the regular expression approach to identify different types of DMs, their representational rules and their poses. Since this paper focuses on the DM defined in Mo *et al*., which is the composition of one or more determinatives together with an optional measure, the DM structure includes prototypical DMs, variant forms of DMs (*e.g*. the ellipsis of the determinative and the insertion of an adjective into a DM[3]), reduplicative forms of DMs (*e.g.* the reduplication of 'M', 'DM' or 'AM'), and forms of the numeral *yi* preceding the reduplicative measures (*e.g.* '*yi*MM' (一 MM) and '*yi*AM' (一 AM)). Due to the infinite of the number of possible DMs, Mo *et al.* [1991] proposed identification of DMs by regular expression before parsing as part of their morphological module in NLP. For example, when the DM compound is the composition of one determinative, *e.g.* numerals in (4), rules (5a), (5b) or (5c) will first apply, and then rules (5d), 5(e) or (5f) will apply to compose complex numeral structures, and finally rules (5g) will apply to generate the pos Neu of numeral structures. From the processes of regular expression, the numerals 534 and 319 in (4) are auto-tagged as Neu.

(4) 鼓勵***534***人完成***319***鄉之旅

guli *wubaisanshisi* *ren* *wancheng*
encourage five hundred thirty four person accomplish
sanbaiyishijiu *xiang* *zhi* *lu*
three hundred and nineteen village DE travel
encourage 534 persons to accomplish the travel around 319 villages

[3] The insertion of an adjective into a DM has the form of '*yi*AM'.

(5) a. NO1 = {〇,一,二,兩,三,四,五,六,七,八,九,十,廿,卅,百,千,萬,億,兆,零,幾};
b. NO2 = {壹,貳,參,肆,伍,陸,柒,捌,玖,拾,佰,仟,萬,億,兆,零,幾};
c. NO3 = {１,２,３,４,５,６,７,８,９,０,百,千,萬,億,兆};
d. IN1 -> { NO1*, NO3*} ;
e. IN2 -> NO2* ;
f. IN3 -> {IN1,IN2} {多,餘,來,幾} ({萬,億,兆}) ;
g. Neu -> {IN1,IN2,IN3,IN4,IN5,DN} ;

Regular expression approach also applies in dealing with ordinal numbers, decimals, and fractional numbers. The ordinal number *diyi* in (6) applies rules (9) and (5g) so that it is regarded as a unit and tagged as Neu. Rules (10) and (5g) apply to decimals such as *sandianyi* in (7). Therefore, decimals are viewed as one unit and tagged as Neu. Depending upon the forms of fractional numbers, rules (11a) or (11b) apply to fractional numbers like *sanfenzhiyi* in (8) and are treated as single units. Then, after application of rules (11a) or (11b), rule (11c) applies to the fractional numbers. Thus the fractional numbers are tagged as Neqa.

(6) 假學術活動中心***第一***會議室召開

jia xveshu huodung zhongxin diyi hueiyishi zhaokai
at academic activity center first auditorium convene
convene at the first auditorium in the Center for Academic Activities

(7) 得分只有***三點一***

defen zhiyou sandianyi
point only three point one
get only 3.1 points

(8) 等於是一日的***三分之一***

dengyu shi yiri de sanfenzhiyi
equal SHI one day DE one third
is equal to one day of third

(9) IN5 -> {第} {IN1,IN2} ;

(10) DN -> IN1 {・, . ,:, ・ ,點} IN1 ;

(11) a. FN1 -> (IN1 {又}) IN1 {分之,一, / } {IN1,DN} ({強,弱}) ;
b. FN2 -> {DN,IN1} {%};
c. Neqa -> {NE5a,WQ,QQ,DQ1,DQ2,PQ,FN1,FN2,FN3,NOP3,RD13} ;

Below, DM structures of classes are also rule design and poses are identified by rule types.

(12) 北市文昌國小*五年一班*

*a. *beishi* *wenchang* *guoxiao* *wunian* *yiban*
Taipei City Wen-Chang elementary school five-M one-M
b. *beishi* *wenchang* *guoxiao* *wunianyiban*
Taipei City Wen-Chang elementary school Fifth Grade Class One
the Fifth Grade Class One in Taipei Wen Chang Elementary school

(13) a. CNP -> IN1 {年} {IN1,ON} {班} ;
b. Ncb -> {NC1,NC2,NC3,CNP,DSP1} ;

DM rules will generate/identify ambiguous DM compounds, *e.g*. (12). If *nian* and *ban* are regarded as measures, *wunianyiban* is segmented into two DMs, *i.e*. *wunian* 'five years' and *yiban* 'one run', like (12a). Therefore, when identifying DMs treated as classes, one has to apply the resolution principles and two DM rules (13a) and (13b). Then, classes in schools such as (12b) are viewed as a single unit, and the noun phrase *wunianyiban* is restricted to be a unit with the pos Ncb. The word segmentation algorithm will reduce semantic anomalies resulting from possible parses of sentences.

When dealing with addresses, especially indicating the floor, number, alley, lane, section and neighbourhood, we also adopt a regular expression approach for identification. The following instances show the same forms with different segmentation between DMs and addresses.

(14) 日前遷至台北市信義路*三段七號三樓之一*

riqian qian zhi taibeishi xinyilu sanduan qihao sanlouzhiyi
a few days ago move to Taipei City XinYi Rd. Sec. 3 No. 7 3F-1
a few days ago, moved to 3F-1, No. 7, XinYi Rd., Sec. 3, Taipei

(15) 行經屏市長安里竹圍巷*一之一0二號*時

xing jing pingshi changanli zhuweixiang yizhiyilingerhao shi
Go through Pingtung City Changan Village Zhuwei Lane No. 1-102 as
when going through No. 1-102, Zhuwei Lane, Changan Village, Pingtung City

(16) a. NC1 -> IN1 {鄰,巷,弄,段,號,樓};
b. NC2 -> IN1 {樓,號} {之,－} IN1 ;
c. NC3 -> IN1 {之,－} IN1 {號} ;

Normally, DMs such as *sanlouzhiyi* and *yizhiyilingerhao* are segmented into several units. The former is segmented into three units, *i.e. sanlou* 'the third floor', *zhi* 'DE' and *yi* 'one' while the latter is segmented into three units, *i.e. yi* 'one', *zhi* 'DE' and *yilingerhao* 'No. 102'. However, according to CKIP Technical Report 96-01 (1996: 50), the determinative measure structures expressing time and location will be combined together as a unit. The reason why the locative DMs are combined is because the first joint principle of segmentation stipulates that when the meaning of a string of words is not obtained from the composition of these components, this string should be segmented as a unit. Consequently, in (14), DM rule (16a) applies to *sanduan* and *qihao*, and DM rule (16b) applies to *sanlouzhiyi*. In (15), *yizhiyilingerhao* applies DM rule (16c). Then *sanduan*, *qihao*, *sanlouzhiyi* and *yizhiyilingerhao* are all processed by the application of DM rule (13b). Thus *sanduan*, *qihao* and *sanlouzhiyi* in (14) and *yizhiyilingerhao* in (15) are all segmented as a single unit and tagged as Ncb, not DM, in Sinica Treebank.

To deal with the reduplicative measures, *e.g.* '*yi*MM' (一 MM) and 'MM' (MM), we also adopt a context-sensitive regular expression to identify them. For example, *zhongzhong* in (17) and *yizhangzhang* in (18) are regarded as a single unit. First, the context-sensitive DM rule (19) is applied, where two measure words in MM are restricted to be equal, and then rule (11c), so the form of reduplicative measures with a preceding optional *yi* is tagged as Neqa.

(17) ***種種***問題

zhongzhong wenti

all kinds of question

all sorts of questions

(18) ***一張張***海報

yizhangzhang haibao

sheets of poster

sheets of posters

(19) RD13 -> ({一}) M M ;

From the examples and rules illustrated above, one knows that the regular expression approach helps people identify certain DMs. However, DMs still have ambiguities.

4. Ambiguities of DMs

The adoption of DMs rules really improves the recall of recognition, but one still has to resolve segmentation, sense, and pos ambiguities of DMs as shown in the preceding examples in Section 1.

4.1 Word Segmentation Ambiguities of DMs

There are two types of word segmentation ambiguities, *i.e*. covering ambiguities and overlapping ambiguities.

Type1: Covering ambiguities

(20) ***一服***藥就見效

a. *yi fuyao jiu jianxiao*

one take medicine then take effect

Every time when he takes medicine, the illness is completely cured.

b. *yifu yao jiu jianxiao*

one-M medicine then take effect

One dose is effective.

(21) 他們家*四口*人

*a. *tamen jia si kou ren*
they family four mouth person

b. *tamen jia sikou ren*
they family four-M person
Their family has four members.

Covering ambiguities are always associated with sense ambiguities, since different segmentations result in different sense interpretations. Goh *et al*. [2005] mention that the covering ambiguity is defined as follows: For a string w = xy, x ∈ W, y ∈ W, and w ∈ W. As almost any single character in Chinese can be considered a word, the above definition reflects only those cases where both word boundaries .../xy/... and .../x/y/... can be found in sentences. Example (20) is ambiguous in its meaning and has two different segmentations. When *fu* functions as a verb, *fuyao* is segmented as a unit, and the meaning of (20) is (20a). When *fu* functions as a measure, *yifu* is tagged as DM, and the meaning of (20) is (20b). Because the combinations of determinatives and measures are countless, DMs such as *yifu* won't be listed in the CKIP dictionary. Different word segmentation will bring structural ambiguities forth. Another segmentation ambiguity exists in the ellipsis of the determinatives. In (21), *kou* is sense ambiguous in that it functions as a noun or as a measure. When *kou* behaves as a noun in (21a), the sentence is semantically anomalous and syntactically ungrammatical. Only when *kou* functions as a measure, will the sentence (21b) be well-processed.

Type 2: Overlapping ambiguities

(22) *一串*串珠飾品

a. *yichuan chuanzhu shipin*
one-M beads accouterment
one string of beads

b. *yichuanchuan zhushipin*
several strings beady crystal
several strings of beady crystals

(23) 媽媽講了*一個*月亮的故事

a. *mama jiang le yige yueliang de gushi*
mother tell LE one-M moon DE story
Mother told a story about the moon.

b. *mama jiang le yige yue liang de gushi*
mother tell LE one-M month Liang DE story
Mother had told for a month about Liang's story.

(24) *第*一派對分三部分

a. *diyi paiduei fen san bufen*
first party divide three part
The first party is divided into three sections.

b. *diyipai dueifen san bufen*
first group dichotomize three part
The first group dichotomized into three parts.

Goh *et al.* [2005] state that overlapping the ambiguity is defined as follows: For a string w = xyz, both w1 = xy ∈ W and w2 = yz ∈ W hold. Mo *et al*. [1991] list a resolution principle to reduce word segmentation ambiguity. The principle asserts if ambiguous word breaks occur between the words in the lexicon and the DMs, the words in the lexicon should have higher priority to get the shared characters. Therefore, (22a), (23a) and (24a) have higher priority to (22b), (23b) and (24b), individually.

According to the above discussions, to resolve word segmentation ambiguities, we propose the following resolution principles which were implemented in the word segmentation system of Ma and Chen [2003].

a) DM compounds are expressed and matched by regular expressions.
b) Lexical words have higher precedence than DM compounds (cf. 22, 23, 24).
c) Longest matching principle (*i.e*. 9, 10, 11, 13, 16, 19): a long DM has higher precedence than a short DM (*cf.* 6, 7, 8, 12 14, 15, 17, 18).
d) Covering ambiguities are resolved by collocation context (cf. 20, 21).

The word segmentation ambiguity is caused by different possible segmentations. Although the above examples have ambiguities, after the application of the resolution principles, the ambiguous segmentation is resolved and the correct segmentation has higher

priority.

4.2 Sense Ambiguities of DMs

Senses and semantic functions of DMs are sometimes related to the types of measures. The sense of certain types of DMs can be identified by types of measures. However, as usual, some DMs have ambiguous senses. Their ambiguity resolution is almost equivalent to word sense disambiguation. Therefore, context sensitive rules and collocation bi-grams are suitable information for resolving sense ambiguities. Methods for word sense disambiguation are also applicable for DMs.

As mentioned in Section 3, we have adopted a regular expression approach for identifying structures denoting addresses. For example, DM rule (16a) is applied to segment *yiduan*, *bahao*, and *yilou* in (25) as single units, and each of them is tagged as Ncb. However, in (26), *hao* and *duan* are both measures specifying the fixed amount or quantity of the road, so *yiyiqihao* and *yiduan* are both tagged as DM. Examples (25) and (26) have sense ambiguities during the processes of DM recognition, so they will be automatically segmented as (25a) and (25b) as well as (26a) and (26b), individually. According to CKIP Technical Report 96-01 (1996), locative DMs are combined so that one can disambiguate (25) and (26). Then, the correct segmentations (25b) and (26b) are resumed according to their contexts.

(25) 羅斯福路*一段八號一樓*

*a. *luosifulu* *yiduan* *bahao* *yilou*
Roosevelt Rd. one-M eight-M first floor

b. *luosifulu* *yiduan* *bahao* *yilou*
Roosevelt Rd. Sec. 1 No. 8 first floor
1 F, No. 8, Roosevelt Rd., Sec. 1

(26) 屬於*１１７號*公路的*一段*

*a. *shuyu* *yiyiqihao* *gonglu* *de* *yiduan*
belong No. 117 road DE Sec. 1

b. *shuyu* *yiyiqihao* *gonglu* *de* *yiduan*
belong 117-M road DE one-M
belong to a part of the 117th road

Similar to dealing with addresses, sense ambiguities occur when one refers to percentages. One can adopt two forms to represent percentage, *i.e.* Chinese characters in (8) and mathematical symbols in (27). Regular expression approach can help one identify word segmentation ambiguity existing in (8). The form of the mathematical symbols has sense ambiguity, which can refer to either percentage like (27) or a time point like (28). Without any context, the symbol "１０／２１" can be identified either as a fractional number and read as "*ershiyifenzhishi*" (二十一分之十 'ten over twenty one') with the pos Neqa or as a time point and read as "*shiyueershiyiri*" (10 月 21 日 'Oct. 21') with the pos Ndabd whose semantic role is time. Besides, the form of mathematical symbols is also used to refer to another kind of time point, such as (29). To reduce such kinds of sense ambiguities caused by mathematical symbols, DM rules (30a), (30b), (31a) and (31b) exist. DM rules (30a) and (30b) apply to the forms of mathematic symbols like (27) tagged as Neqa (numbers), while DM rules (31a) and (31b) apply to forms like (29) tagged as Nd (time point). Although DM rules (30) and (31) help disambiguate sense ambiguities between forms of mathematic symbols denoting percentage and time points such as (27) and (29), one still has to have context to make (27) and (28) distinguishable.

(27) ４０分的佔了***２／３***

sishifen de zhan le sanfenzhier

40-M DE occupy LE two-thirds

Those of forty points occupy two-thirds.

(28) ***１０／２１***召開全院網路工作小組第三次會議

shiyuershiyiri zhaokai quan yuan wanglu gongzuo xiaozu disanci huiyi

Oct. 21 convene whole faculty network work group third conference

convene the third conference of the network group on Oct. 21

(29) ***2005／06／30***更新

2005／06／30 gengxin

June 30, 2005 update

update on June 30, 2005

(30) a. NE5a -> {NE2} {—,／} {NE2} ;

b. Neqa -> {NE5a,WQ,QQ,DQ1,DQ2,PQ,FN1,FN2,FN3,NOP3,RD13} ;

(31) a. NE5b -> {NE2} {—,／} {NE2} {—,／} {NE2} ;

b. Nd -> {Ndabe,NE5b,ND3,ND5}

Take Chao's [1968] example to help disambiguate ambiguities similar to those between (27) and (28). As Chao discusses, the form *Guangxu sanshisinian* (光緒三十四年) can be either the phrase 'the thirty-fourth year of Guangxu (*i.e.*, 1908)' or the sentence 'Guangxu's reign was thirty-four years [long].' Chao believes that, in most cases, the context will resolve the ambiguity. The following examples in Sinica Corpus have similar ambiguities to those Chao mentions.

(32) 經過卡斯楚*三十年*的統治之後

jingguo kasichu sanshinian de tongzhi zhihou

after Castro 30-M DE governance afterward

after Castro's thirty-year governance

(33) *三十年*秋，緝私總隊復正名爲稅警總團

sanshinian qiu qisi zongdui fu zhengmingwei shuijingzongtuan

the year of thirty autumn anti-smuggling team again rectify tax policemen team

In the autumn in the year of thirty, the anti-smuggling team is rectified to the tax policemen team.

Examples (32) and (33) have the same temporal phrase *sanshinian*, but their senses, functions and roles are different. The temporal phrase in (32) denotes a time length and is tagged as DM. The semantic role of *sanshinian* in (32) is duration. However, *sanshinian* in (33) indicates a time point and is tagged as Ndaad, whose semantic role is time. Even though example (33) omits either a Chinese reign title or *Mingguo* (民國) preceding *sanshinian*, we still know *sanshinian* is a specific time, not a period, from the context. When the measures *nian* 'year' (年) and *ri* 'day' (日) are preceded by numerals, the temporal phrases always have sense ambiguities. We have two tricks to differentiate between time points and time lengths. If DMs

denote time points, they are usually preceded by key words of *Mingguo*, the Christian era like *Gongyuan* (公元) and *Xiyuan* (西元), or a Chinese reign title such as *Guanxu* (光緒), *Qianlong* (乾隆), *Tianbao* (天寶), *Jiajing* (嘉靖) and so on. When DMs are preceded by these key words, they are tagged as Ndaad (*i.e*. a time point). Another trick that helps one to recognize DMs as time points is their neighbouring temporal phrases. Time points, not time lengths, are usually combined with two or three co-occurring temporal phrases, *e.g*. *erlinglingwunian liuyue* (2005 年 6 月 'June 2005'), *liuyue sanshiri* (6 月 30 日 'June 30'), *erlinglingwunian liuyue sanshiri* (2005 年 6 月 30 日 'June 30, 2005'), etc. The two tricks mentioned above and context will help one reduce some ambiguities of phrases and mathematical symbols specifying time.

In conclusion, sense ambiguity resolution of DMs is almost equivalent to word sense disambiguation. Therefore, context sensitive rules and collocation bi-grams are suitable information for resolving sense ambiguities. Methods for word sense disambiguation are also applicable here.

4.3 POS Ambiguities of DMs

Here, we first discuss ambiguities about temporal adverbs to illustrate pos ambiguities and possible resolution methods. The temporal phrases in (34) and (35) are regarded either as time points or as time lengths. These temporal phrases have the same strings and word segmentation but have different parts of speech, semantic roles and functions. As is known, in Chinese a folktale, a woman called *Wang Baochuan* (王寶釧) went through 18 years of hardship for her husband's turning back home. In Mainland China, the Tiananmen Square massacre occurred in 1989. Therefore, the semantic roles of the temporal phrases *shibanian* in (34) and *bajiunian* in (35) will be labelled as duration and time individually. The pos of the former is DM while that of the latter is Ndaad. The reason for making different assignments of semantic roles is concerned with logical interpretation of sense collocations according to common sense and the real world knowledge.

(34) 18年的苦守

shibanian de kushou

18-M DE wait bitter

wait bitter for eighteen years

(35) 89年的反抗

bajiunian *de* *fankang*

the year of 89 DE revolt

the revolt in the year of 89

When detecting DMs in Sinica Corpus and Sinica Treebank, one finds some interesting examples. The verb phrases (36) and (37) have the same lexical items except for their linear word order. The pos of the DM structure in (36) is DM whose semantic role is duration while that in (37) is Ndaad whose semantic role is time. It seems that different positions of temporal DMs will affect the meanings of sentences. Therefore, we briefly reviewed the data in Sinica Treebank. The totality of the semantic role time of NPs and of PPs following a verb is close to that of the semantic role duration. But the totality of the semantic role time of NPs and of PPs preceding a verb is much more than that of duration. The statistics indicate that temporal DMs preceding verbs mostly function as time. Another problem with assignment of semantic role to a similar structure is illustrated by (38) and (39). The DM structure in (38) is tagged as DM and assigned the semantic role duration while, in (39), it is tagged as Ndaad and assigned the semantic role time. This kind of pos ambiguity has relation to situation types. The situation type of *fuxing* in (38) is an Activity while that of *panxing* in (39) is an Achievement. The feature [±Durative][4] of the events causes differences. As for the pos ambiguity of *yixi*a, *yixia* in (40) means 'for a while', which is tagged as Nddc and assigned the role of duration. However, *yixia* in (41) means 'once', which is tagged as DM and assigned the role of frequency. Nevertheless, *yixia* in (42) is POS ambiguous and has two senses. One is tagged as Nddc and means 'for a while' while another is tagged as DM and means 'once'. The former is labelled as duration and the latter is assigned as frequency. Equal to the cause of differences between (38) and (39), the ambiguities in (42) are due to situation types.

(36) 親政三十八年

qinzheng *sanshibanian*

hole the reins of government 38-M

hold the reins of government for 38 years

[4] More detailed discussion about situation types can be referred to Smith [1991].

(37) 三十八年親政

sanshibanian *qinzheng*

the year of 38 hold the reins of government

hold the reins of government in the year of 38

(38) 34年的服刑

sanshisinian *de* *fuxing*

34-M DE serve a sentence

serve a sentence for 34 years

(39) 34年的判刑

sanshisinian *de* *panxing*

the year of 34 DE sentence

sentence a person in the year of 34

(40) 等我一下

deng wo *yixia*

wait I for a while

wait for me for a while

(41) 敲他一下

Qiao *ta* *yixia*

strike he one-M

strike him once

(42) 咬他一下

a. *yao* *ta* *yixia*

bite he for a while

bite him for a while

b. *yao* *ta* *yixia*

bite he one-M

bite him once

Semantic role assignment is not an easy task, since it requires world knowledge as well as linguistic knowledge. In You and Chen [2004], they identify parameters of determining semantic roles and propose an instance-based approach to resolve ambiguities. They adopt dependency decision making and example-based approaches. Semantic roles are determined by four parameters, including syntactic and semantic categories of the target word, case markers, phrasal head, and sub-categorization frame and its syntactic patterns. The refinements of features extraction, canonical representation for certain classes of words and dependency decisions improve role assignment. To assign the semantic roles of DMs, the above parameters are further refined as the features of relative positions and situation types.

The examples above show that ambiguities are unavoidable when one deals with DMs. In addition to the typical DMs, some related structures like reduplicative DMs, numerals, the ellipsis of measures, etc. are also topics for discussion. During DM processing, certain DMs are ambiguous to automatic identification in word segmentations, senses as well as poses. Here, *yi dian* (一點) is taken as an example.

(43) 有一點要特別注意

you yidian yau tebie zhuyi
have one-M should special attention
There is a point very important for attention.

(44) 一點心意你要收下

yidian xinyi ni yao shouxia
little regard you should receive
You must receive my little thanks.

(45) 一點集合

yidian jihe
one o'clock assemble
assemble at one o'clock

(46) 漂亮一點

piaoliang yidian
beautiful a little
a little bit beautiful

(47) 快一點

a. *kuai* *yidian*
nearly one o'clock
nearly one o'clock

b. *kuai* *yidian*
fast a little
more quickly

(48) 慢一點

man *yidian*
slow a little
more slowly

The phrase *yidian* functions as a pronoun and tagged as DM in (43), functions as a quantitative determinative modifying *xinyi* and tagged as Neqa in (44), functions as a time noun and tagged as Ndabe in (45), and functions as a post-verb adverb of degree and tagged as Dfb in (46). While in (47), *yidian* has sense ambiguities depending upon context. In addition, *yidian* in (47a) and (48) is pos ambiguous. For another example, *qi* (起) functions as a measure in both *siqi anjian* (四起案件) and *yiqi mingan* (一起命案). In *fongyun siqi* (風雲四起) and *yiqi sikao* (一起思考), *siqi* and *yiqi* are tagged as VA11 and Dh individually. However, in *yiqi fanan* (一起犯案) and *fanan siqi* (犯案四起), the DMs *yiqi* and *siqi* are ambiguous. It is obvious that the ambiguities of DMs are complex and that a DM compound can have more than one classification of ambiguities.

No matter whether the ambiguity is from word segmentation, sense, or pos, the prescription of resolution principles and DM rules are helpful in disambiguating DMs. Besides, the neighbouring morphemes and context are other tricks in reducing ambiguities. Furthermore, pos ambiguities are concerned with common sense, and the resolution features also include positions of temporal DMs and situation types. Such ambiguities have to be reduced by the application of parameters of context vector models.

5. Implementation and Evaluation

We randomly chose 2035 sentences (11697 word tokens) from Sinica Treebank as our development set. In total, 545 tokens of the development data are processed by the revised DM rules (as shown in the appendix). Among the 545 tokens, 504 tokens are correctly

segmented, and 443 tokens are correctly pos tagged. The segmentation accuracy of the development data is 92.5%, the tag accuracy of the development data is 81.3%, and the tag accuracy with the correct segmentation of the development data is 88.0%. Contrastively, the segmentation accuracy and tag accuracy of the development set processed by the original DM rules are both lower than those applied the revised DM rules. The segmentation accuracy is 84.2%, and the tag accuracy is 71.0%. Then, to test the accuracy of the revised DM rules, we randomly chose 2111 sentences (12209 word tokens), which have no overlap with the development set, from Sinica Treebank as the testing set. In total 564 tokens of the testing data were processed using the revised DM rules. Among those 564 tokens, 508 tokens were correctly segmented, and 424 tokens were correctly pos tagged. By application of the revised DM rules, the segmentation accuracy of the testing data is 90.1%, the tag accuracy of the testing data is 75.2%, and the tag accuracy with the correct segmentation of the testing data is 83.5%. Contrastively, processed by the original DM rules, the segmentation accuracy and the tag accuracy of the testing data is 77.8% and 60.3% individually. Table 1 is the evaluation result.

Table 1. Accuracy of Development data and Testing data

accuracy \ data		development set	testing set
		2035 sentences (11697 word tokens)	2111 sentences (12209 word tokens)
original DM rules	segmentation accuracy	84.2%	77.8%
	tag accuracy	71.0%	60.3%
revised DM rules	segmentation accuracy	92.5%	90.1%
	tag accuracy	81.3%	75.2%
	tag accuracy with correct segmentation	88.0%	83.5%

Table 1 shows that both segmentation accuracy and tag accuracy of development set and of testing set processed by the revised DM rules are higher than those processed by the original DM rules, although the segmentation accuracy, tag accuracy and tag accuracy with correct segmentation of the testing set are a little bit lower than those of the development set.

After data analysis, we found that there were several reasons for inaccurate results. The most crucial factor resulting in inaccuracy is ambiguity, including word segmentation ambiguities, sense ambiguities and pos ambiguities. The word segmentation ambiguities, sense ambiguities, and pos ambiguities caused 11%, 32.4% and 27.2% of errors, respectively. For example, (49a) and (50a) are correct sentences. The contrastive sentences (49b) and (50b)

have errors in word segmentation ambiguities. Sentences (51b) and (52b) have errors in sense ambiguities, which are contrastive to the correct ones (51a) and (52a). Sentences (53b) and (54b) have errors in pos ambiguities. The total percentage of errors caused by ambiguities is about 70.6%.

(49) a. 汽車(Na) 可(D) 不(D) 是(SHI) ***一個(DM)*** 人(Na) 所(D) 發明(VC) 的(T)

qiche ke bu shi yige ren suo faming de

automobile should NEG SHI one-M person that which invent DE

Automobiles are not invented by one person.

*b. 汽車(Na) 可(D) 不(D) 是(SHI) 一(Neu) 個人(Nh) 所(D) 發明(VC) 的(DE)

qiche ke bu shi yi geren suo faming de

automobile should NEG SHI one individual that which invent DE

(50) a. ***有一次(DM)*** 要(VE) 刨(VC) 木(Na) 時(Ng)

youyici you pau mu shi

once should shave wood as

once when shaving wood

*b. 有(V_2) 一(Neu) 次要(A) 刨木(Na) 時(Ng)

you yi ciyou pau mu shi

have one secondary shave wood as

(51) a. 都(D) 有(V_2) ***一段(DM)*** 感人(VH) 的(DE) 故事(Na)

dou you yiduan ganren de gushi

all have one-M heart-stirring DE story

all have one heart-stirring story

*b. 都(D) 有(V_2) 一段(Nc) 感人(VH) 的(DE) 故事(Na)

dou you yiduan ganren de gushi

all have Sec. 1 heart-stirring DE story

(52) a. 但(Cbb) 也(D) 付出(VC) ***數位(DM)*** 創業(Nv) 先進(Na) 寶貴(VH) 的(DE) 生命(Na) 及(Caa) 損失(VJ) １４架(DM) 飛機(Na) 的(DE) 慘痛(VH) 代價(Na)

dan ye fuchu shuwei chuangye xianjin baoguei de shengming
but also pay several-M pioneering precursor valued DE life
ji sunshi shisijia feiji de cantong daijia
and lose 14-M airplane DE cruel cost
but also pay several pioneering precursors' valued lives and suffer the cruel costs of losing fourteen airplanes

*b.但(Cbb) 也(D) 付出(VC) 數位(A) 創業(VA) 先進(VH) 寶貴(VH) 的(DE) 生命(Na) 及(Caa) 損失(Na) １４架(DM) 飛機(Na) 的(DE) 慘痛(VH) 代價(Na)

dan ye fuchu shuwei chuangye xianjin baoguei de shengming
but also pay several-M digital precursor valued DE life
ji sunshi shisijia feiji de cantong daijia
and lose 14-M airplane DE cruel cost

(53) a. 10月(Nd) ***5日(Nd)***

shiyue wuri
Oct. fifth
Oct. 5

*b. 10月(Nd) 5日(DM)

shiyue wuri
Oct. five-M

(54) a. 對於(P) ***七十九年(Nd)*** 年終(Nd) 獎金(Na)

dueiyu qishijiunian nianzhong jiangjin
about the year of 79 year-end bonus
about the year-end bonus in the year of 79

*b. 對於(P) 七十九年(DM) 年終(Nd) 獎金(Na)

dueiyu qishijiunian nianzhong jiangjin
about 79-M year-end bonus

Other than errors in ambiguities, there are four kinds of errors bringing about inaccuracy. The first kind of errors is a result of the segmentation model (a Hidden Markov Model). In HMM, there are several possible paths, including the correct one. However, the result chosen was not the correct one. For example, (55a) and (55b) are the possible paths in HMM. For the result, the inaccurate one (55b) was chosen. The error of (56b) is also due to HMM. The percentage of errors made by HMM is 10.3%.

(55) a. 回想(VE) 起(Di) ***二十年(DM)*** ***前(Ng)*** 的(DE) 往事(Na)

hueixiang qi ershinian qian de wangshi

recall ASP 20-M before DE past

recall the past twenty years ago

*b. 回想起(VE) 二十(Neu) 年(Nf) 前(Ng) 的(DE) 往事(Na)

hueixiangqi ershi nian qian de wangshi

recall twenty M before DE past

(56) a. ***六十六歲(DM)*** 時(Ng)

liushiliusuei shi

66 years old as

as 66 years old

*b. 六十六(Neu) 歲(Nf) 時(Ng)

liushiliu suei shi

66 M as

The second kind of errors is because different contexts cause different tagging. In Sinica Treebank, one or more determinatives together with an optional measure will constitute a DM. However, certain determinatives and measures are tagged differently than usual because of the context. In (57a), *liangsanmiaozhong* (兩三秒鐘 'two and three seconds') is composed of *liangmiaozhong* 'two seconds' and *sanmiaozhong* 'three seconds'. The measure *miaozhong* 'second' is shared by determinatives *liang* 'two' and *san* 'three'. What is in (57c) is the tree structure of (57a). Both *liang* and *san* are tagged as Neu, and *miaozhong* is tagged as Nf. In (58a), *qibayue* (七八月 'July and August') is composed of *qiyue* 'July' and *bayue* 'August'. The diagram (58c) is the bracketed tree diagram of (58a). From the context, one knows that *qi* in (58a) is not the numeral 'seven' but a month 'July' so *qi* is tagged as Nd not Neu. The

percentage of errors resulting from contexts is 8.8%.

(57) a. 這(Nep) 姿勢(Na) 保持(VJ) ***兩(Neu) 三(Neu) 秒鐘(Nf)***

zhe zishi baochi liang san miaozhong

this pose keep two three M

This pose is keeping for two and three seconds.

*b. 這(Nep) 姿勢(Na) 保持(VJ) 兩三秒鐘(DM)

zhe zishi baochi liangsanmiaozhong

this pose keep two-three-M

c. S(theme:NP(quantifier:Nep:這|Head:Nac:姿勢)|Head:VJ1:保持
|duration:DM(Head:Neu(Head:Neu:兩|Head:Neu:三)|Head:Nfg:秒鐘))

(58) a. ***7(Nd)*** 、(Caa) ***8月(Nd)*** 爲(VG) 下午(Nd) ４點(Nd) 至(Caa) ４ 點(Nd) ３０分(Nd)

qi bayue wei xiawu sidian zhi sidian sanshifen

seven August is afternoon four o'clock to four o'clock thirty minutes

In July and August, it is from four to four-thirty o'clock.

*b. ７(Neu) 、(PAUSECATEGORY) ８月(Nd) 爲(P) 下午(Nd) ４點(DM) 至(Caa) ４點(DM) ３０分(DM)

qi bayue wei xiawu sidian zhi sidian sanshifen

seven August is afternoon four-M to four-M thirty minutes

c. S(theme:NP(Head:Ndabc(DUMMY1:Ndabc:７|Head:Caa:、
|DUMMY2:Ndabc:8月))|Head:VG2:爲|range:NP(property:Ndabe:下午
|Head:NP(DUMMY1:NP(Head:Ndabe: ４點)|Head:Caa:至
|DUMMY2:NP(property:Ndabe:４點|Head:Ndabe:３０分))))

The third kind of errors occurs when there is only one measure without any other determinatives, *e.g.* (59b) and (60b). The percentage of errors in one measure is 8.1%. The error in wrong tagging of one measure is because of the training data from Sinica Corpus. A sole measure is tagged as Nf in Sinica Corpus, but in Sinica Treebank it is viewed as a DM structure and tagged as DM. Therefore, a sole measure always has incorrect pos. This kind of error has to be dealt with during postprocessing.

(59) a. 最近(Nd) 聽到(VE) 了(Di) ***個(DM)*** 駭人聽聞(VH) 的(DE) 故事(Na)

zueijin tingdao le ge hairentingwen de gushi

recent hear LE M shocking DE story

recently hear one shocking story

*b. 最近(Nd) 聽到(VE) 了(Di) 個(Nf) 駭人聽聞(VH) 的(DE) 故事(Na)

zueijin tingdao le ge hairentingwen de gushi

recent hear LE M shocking DE story

(60) a. 有(V_2) ***顆(DM)*** 善良(VH) 的(DE) 心(Na)

you ke shanliang de xin

have M kindhearted DE heart

is kindhearted

*b. 有(V_2) 顆(Nf) 善良(VH) 的(DE) 心(Na)

you ke shanliang de xin

have M kindhearted DE heart

The last kind of error is because of unknown word identification such as (61) and (62). The unknown words in (61b) and (62b) are not identified correctly, so errors occur. The percentage of errors in unknown words is 2.2%.

(61) a. 誰(Nh) 言(VE) 寸(DM) 草(Na) 心(Na)

shei yan cun cao xin

who say inch grass heart

children like grass (cannot pay their parents back)

*b. 誰(Nh) 言(VE) 寸草(Na) 心(Na)

shei yan cuncao xin

who say grass heart

(62) a. 報(VC) 得(DE) 三(Neu) 春(Nd) 暉(Na)

bao de san chun huei

pay back DE three spring sunshine

pay parents back

*b. 報得三春暉(VH)

baodesanchunhuei

pay back

In our implementation and evaluation, the accuracy of segmentation, of tag, and of tag with correct segmentation of the development set processed by the revised DM rules are higher than those processed by the original DM rules. Through application of the revised DM rules, the segmentation accuracy, tag accuracy and tag accuracy with correct segmentation of the testing set are a little bit lower than those of the development set. The percentage of ambiguities causing inaccuracy is 70.6% while the total percentage of other factors is 29.4%. The high proportion of ambiguity shows that, although a regular expression approach was used in applying DM rules to deal with DMs, eventually, ambiguity is the most crucial issue one must confront. Therefore, the application of resolution principles, of DM rules, of context sensitive rules, of collocation bi-grams and of parameters of context vector models are necessary to help one disambiguate. Language reflects the human view of the world. Differing personal world knowledge may result in different explanations of sentences. Some reduction of ambiguities of DMs depends upon the human's common sense knowledge.

6. Conclusion

DMs are not a closed set, so one has to apply DM rules during the process of automatic identification of DMs. By observing Sinica Treebank, we had developed a set of regular expression rules to identify DMs and their parts of speech. Thus, all DM candidates can be matched and classified by regular expression rules. However, due to segmentation, pos and sense ambiguities of DMs, DM rules are necessary complements to dictionaries and helpful to resolve ambiguities by applying resolution principles and segmentation models. Sense and pos ambiguities are also expected to be resolved by different approaches during post-processing by applying context sensitive rules, collocation bi-grams and parameters of context vector models.

References

中央研究院詞庫小組(CKIP), "「搜」文解字—中文詞界研究與資訊用分詞標準," 技術報告 96-01, 中央研究院,台北, 1996.

何杰(He, J.), *現代漢語量詞研究*, 民族出版社, 北京市, 2002.

黃居仁, 陳克健, 賴慶雄(編著), *國語日報量詞典*, 國語日報出版社, 台北, 1997.

Academia Sinica, CKIP word segmentation system, http://ckipsvr.iis.sinica.edu.tw/.

Academia Sinica, Sinica Corpus, version 4.0, http://www.sinica.edu.tw/SinicaCorpus/, 2001.

Academia Sinica, Sinica Treebank, version 3.1, http://treebank.sinica.edu.tw/, 2006.

Chao, Y.-R., *A Grammar of Spoken Chines*e, University of California Press, Berkeley, 1968.

Crystal, D., *A Dictionary of Linguistics and Phonetics*, Blackwell, Massachusetts, Cambridge, 1991.

Goh, C.-L., M. Asahara and Y. Matsumoto, "Chinese Word Segmentation by Classification of Characters," *Computational Linguistics and Chinese Language Processing*, 10(3), 2005, pp. 381-96.

Jurafsky, D. and J. H. Martin, *Speech and Language Processing: An Introduction to natural Language Processing, Computational Linguistics, and Speech Recognition*, Prentice Hall, Upper Saddle River, N.J., 2000.

Li, C. N. and S. A. Thompson, *Mandarin Chinese: A Functional Reference Grammar*, University of California Press, Berkeley, 1981.

Ma, W.-Y. and K.-J. Chen, "Introduction to CKIP Chinese Word Segmentation System for the First International Chinese Word Segmentation Bakeoff," In *Proceedings of ACL, Second SIGHAN Workshop on Chinese Language Processing*, 2003, Sapporo, Japan, pp.168-171.

Mo, R.-P. J., Y.-J. Yang, K.-J. Chen and C.-R. Huang, "Determinative-Measure Compounds in Mandarin Chinese: Their Formation Rules and Parser Implementation," In *Proceedings of ROCLING IV (R.O.C. Computational linguistics Conference)*, 1991, National Chiao-Tung University, Hsinchu, Taiwan, pp. 111-134.

Prins, R. P., Finite-State Pre-Processing for Natural Language Analysis, Art Dissertation, 2005

Smith, C. S., *The Parameter of Aspect*, Kluwer Academic Publishers, Dordrecht, 1991.

Tai, J. H-Y, "Chinese classifier systems and human categorization," *In Honor of William S-Y. Wang: Interdisciplinary Studies on Language and Language Change*, ed. by M. Y. Chen and O J.-L. Tzeng, Pyramid Press, Taipei, 1994, pp. 479-494.

You, J.-M. and K.-J. Chen, "Automatic Semantic Role Assignment for a Tree Structure," In *Proceedings of 3rd ACL SIGHAN Workshop*, 2004, Barcelona, Spain, pp. 109-115.

Appendix. CKIP's DM Phrase Structure Rules

Abbreviation	Term
A	adjectives
DD	demonstrative determinatives
DESC	the adjectives that occur in the middle of a DM compound
DFa	pre-verbal degree adverbs
Dfb	post-verbal degree adverbs
Dh	manner adverbs
DM	determinative-measure compounds
DQ	quantitative determinatives denoting degree
DS	definite specific determinatives
M	measure words
NC	place nouns
Ncb	common place nouns
Nd	time nouns
Ndaad	time nouns indicating years
Ndabd	time nouns indicating days that are circular
Nddc	time nouns indicating future
Neqa	quantitative determinatives
Neu	numeral determinatives
Nf	measures
NO	numeral determinatives
ON	ordinal numerical determinatives
OS	ordinal specific determinatives
PNM	post nominal modifiers
PQ	quantitative determinatives denoting par relation
QO	interrogative quantitative determinatives
WQ	quantitative determinatives denoting totality
VA11	active intransitive motion verbs

NO1 = {〇,一,二,兩,三,四,五,六,七,八,九,十,廿,卅,百,千,萬,億,兆,零,幾} ;

NO2 = {壹,貳,參,肆,伍,陸,柒,捌,玖,拾,佰,仟,萬,億,兆,零,幾} ;

NO3 = {１,２,３,４,５,６,７,８,９,０,百,千,萬,億,兆} ;

NO3a = {百,千,佰,仟,萬,億,兆} ;

ON = {甲,乙,丙,丁,戊} ;

NC = {國,省,州,縣,鄉,村,鎮,鄰,里,郡,區,站,巷,弄,段,號,地,街,樓,衖,市,洲,部,司,課,院,科,系,級,股,室,廳} ;

Ndabe = {清晨,凌晨,早晨,早上,晚上,上午,中午,下午,晨間,午間,晚間,半夜,午夜,晨,午,晚,傍晚,深夜,昌午,子時,丑時,寅時,卯時,辰時,巳時,午時,未時,申時,酉時,戌時,亥時} ;

ND = {微秒,釐秒,秒,秒鐘,分,分鐘,刻,刻鐘,點,點鐘,點多鐘,更,旬,紀,輪,天,日,週,周,禮拜,季,年,年份,載,號,晚,宿,週年,周年,周歲,會,會兒,陣,世,輩,年期} ;

ND2 = {時,世紀,年度,月,月份,陣子,學期,學年,年代,下子} ;

DESC = {大,小,整} ;

PNM = {多,餘,半,出頭,好幾,開外,整,正,許,足,之多} ;

Nfa = {本,把,瓣,部,柄,床,處,期,齣,場,朵,頂,堵,道,頓,錠,闋,棟,幢,檔子,檔,封,幅,發,分,份,服,個,根,根兒,管,行,戶,件,家,架,卷,具,節,句,屆,箇,捲,劑,隻,尊,盞,張,枝,支,樁,幀,只,株,折,炷,軸,口,棵,款,客,輛,粒,片,輪,枚,面,門,幕,匹,所,艘,扇,首,乘,襲,頭,條,長條,台,臺,挺,堂,帖,顆,座,則,冊,任,尾,味,位,頁,葉,房,彎,班,員,介,丸,名,項,起,間,篇,題,目,招,股,回} ;

Nfb = {通,口,頓,盤,局,番} ;

Nfc = {對,雙,宗,番,畦,餐,行,身,列,長列,系列,排,長排,副,付,套,筆,串,長串,掛,幫,房,批,組,窩,綑,捆,群,胎,桌,嘟嚕,部,種,類,樣,樣兒,派,路,壟,落,伙,夥,束,簇,席,疊,紮,色,票,叢,隊,攤,式,蓬,項} ;

Nfd = {些,分,分兒,團,堆,泡,綹,撮,把,股,灘,汪,陣,口,口兒,抹,塊,滴,欄,捧,抱,層,重,帶,截,長截,長截兒,截兒,節,節兒,長節,長節兒,段,段兒,長段,長段兒,絲,絲兒,點,點兒,片,縷,部份,部分,坨,匹,疋,階,抔,波,道} ;

Nfe = {盒,盒子,匣,匣子,箱,箱子,櫃子,櫥,櫥子,籃,籃子,簍,簍子,爐子,包,包兒,袋,袋兒,池子,瓶,桶,聽,罐,罈,盆,鍋,籠,盤,碗,杯,勺,勺子,匙,湯匙,筒,擔,瓶子,桶子,罐子,罈子,盆子,鍋子,籠子,盤子,杯子,筒子,擔子,籮筐,杓,杓子,茶匙,壺,盅,筐,瓢,鍬,缸} ;

Nff = {身,頭,臉,鼻子,嘴,肚子,手,腳,桌子,院子,地,屋子,池,腔,家子} ;

Nfg = {公厘,公寸,公分,公尺,公丈,公引,公里,市尺,公釐,營造尺,台尺,吋,呎,碼,哩,浬,海浬,度,噚,尺,里,釐,寸,丈,米,厘,厘米,海哩,海里,英尺,英里,英呎,英寸,米突,米尺,微米,毫米,英吋,英哩,光年,公畝,公頃,市畝,營造畝,坪,畝,分,甲,頃,平方公里,平方公尺,平方尺,平方公分,平方英哩,英畝,公克,公斤,公噸,市斤,台兩,台斤,日斤,盎司,盎斯,磅,公擔,公衡,公兩,克拉,斤,兩,錢,噸,克,英磅,英兩,公錢,毫克,毫分,公毫,仟克,公撮,公升,市升,營造升,台升,日升,盎司,品脫,加侖,蒲式耳,公斗,公石,公秉,公合,公勺,斗,毫升,夸,夸特,夸爾,立方米,立方厘米,立方公分,立方公寸,立方公尺,立方公里,立方英尺,石,斛,西西,角,毛,元,圓,塊,先令,盧比,法郎,法朗,辨士,馬克,鎊,英鎊,美元,便士,里拉,日元,日圓,刀,打,令,綸,籮,大籮,焦耳,千卡,仟卡,燭光,仟瓦,千瓦,伏特,馬力,爾格,瓦特,瓦,卡路里,卡,馬克,仟赫,千赫,兆赫,赫,赫茲,位元,莫耳,歐姆,法拉第,安培,分貝,居里,微居里,毫居里,毫安培,毫米,毫巴,達因,牛頓,周波,歲,℃} ;

Nfh = {程,作,分,厘,毫,絲,圍,指,象限,度,開,開金,聯,師,旅,團,營,伍,班,排,連,球,波,回合,折,階,摺,等,票,流,棒,聲,次,股} ;

Nfi = {度,輪,回,次,遍,趟,下,下兒,遭,番,聲,聲兒,響,圈,圈兒,步,把,仗,覺,頓,關,手,手兒,腳,掌,巴掌,拳,拳頭,眼,口,刀,槌,槌子,板,板子,鞭,鞭子,棒,棍,棍子,陣,針,箭,槍,槍矛,砲,場,周,曲,跤,記,回合,票} ;

M = Nfa & Nfb & Nfc & Nfd & Nfe & Nfg & Nfh & Nfi & ND ;

TPNM = {半,多,許,整,正} ;

WQ = {一,全,滿,整,成,一切,所有} ;

QQ = {多少,若干,幾多} ;

DFa = {很,挺,怪,真,好,極,滿,更,再,頂,最,太,忒,多,夠,非常,異常,十分,尤其,有點,略爲,稍微,比較,不大,過分,過份,這麼,那麼} ;

DQ1 = {多,許多,許許多多,有些,好些,幾許,有的,少許,多數,少數,大多數,泰半,不少,部分,一部分,部份,個把} ;

DQ2 -> DFa {多,少} ;

PQ = {半,若干,有的} ;

DD = {這,那,哪};

OS = {上,下,前,後,頭,末,次,首,某,另,同} ;

DS = {本,貴,敝,什麼,啥,諸,何,別,旁} ;

IN1 -> { NO1*,NO3* } ;

IN2 -> NO2* ;

IN3 -> {IN1,IN2} {多,餘,來,幾} ({萬,億,兆}) ;

IN3a -> NO3a* ;

IN4 -> {上} IN3a ;

IN5 -> {第} {IN1,IN2} ;

DN -> IN1 {・, . ,:, ・,點} IN1 ;

NE5a -> {IN2} {—,／} {IN2} ;

NE5b -> {IN2} {—,／} {IN2} {—,／} {IN2} ;

FN1 -> (IN1 {又}) IN1 {分之,—,／} {IN1,DN} ({強,弱}) ;

FN2 -> {DN,IN1} {%} ;

FN3 -> {IN1} {成} ({IN1,PNM}) ;

FN = FN1, FN2, FN3 ;

NA1 -> IN1 {年級} ;

NC1 -> IN1 {鄰,巷,弄,段,號,樓} ;

NC2 -> IN1 {樓,號} {之,－} IN1 ;

NC3 -> IN1 {之,－} IN1 {號} ;

ND1 -> {IN1,這,那) ND ;

ND3 -> IN1 ND2 ;

ND4 -> IN1 ND (PNM,TPNM);

ND5 -> {這,那} {時,陣子,下子};

ONP -> ON M ;

NOP1 -> IN1 (DESC) ({半}) M ;

NOP2 -> DESC (半) M ;

NOP3 -> IN1 PNM ;

NOP4 -> M (PNM) ;

NOP5 -> {IN3,DN,FN,雙} M ;

NOP -> {FN,NOP1,NOP3,NOP4,NOP5} ;

WQP -> WQ M ;

WQP -> WQ Nff ;

WQP -> {整整,滿滿} NOP1 ;

QQP -> QQ NOP4 ;

DQP1 -> {好幾} {NOP1,NOP2,NOP4} ;

DQP2 -> {DQ1,DQ2} M ;

DQP -> {DQP1,DQP2} ;

PQP1 -> {數} {NOP1,NOP2,NOP4} ;

PQP2 -> PQ NOP4 ;

PQP -> {PQP1,PQP2} ;

XQP -> {WQP,QQP,DQP,PQP} ;

CNP -> IN1 {年} {IN1,ON} {班} ;

DSP1 -> {他} {國,省,州,縣,鄉,村,鎮,鄰,里,郡,區,站,巷,弄,段,號,地,樓,衖,市,洲} ;

DSP2 -> {該} {NOP,PQP} ;

DSP3 -> DS M ;

DSP -> {DSP1,DSP2} ;

OSP1 -> {第} NOP1 ;

OSP2 -> {每} {XQP,NOP,DSP2} ;

OSP2 -> {各} {XQP,NOP,DSP2} ;

OSP2 -> {逐} M ;

OSP3 -> {另外,近,將近} {PQP,NOP1,NOP5} ;

OSP4 -> OS {NOP,PQP} ;

DDP1 -> DD {WQP,DQP,PQP,NOP,NOP2} ;

DDP2 -> {此} {OSP1,NOP} ;

OSP -> {OSP1,OSP2,OSP4} ;

OHSP -> ({其它,其他,其餘}) {任何} {NOP1,DSP} ;

HOSP -> ({任何}) {其它,其他,其餘} {XQP,DDP1,OSP,NOP,ONP} ;

STDM -> IN1 {秒} IN1 ;

RNOP1 -> IN1 (DESC) M ;

RNOP2 -> {半} M ;

RNOP3 -> {DESC,成} M ;

RD13 -> ({一}) M M ;

Nac -> {NA1} ;

Ncb -> {NC1,NC2,NC3,CNP,DSP1} ;

Neqa -> {WQ,QQ,DQ1,DQ2,PQ,FN1,FN2,FN3,NOP3,RD13} ;

Neqb -> {PNM,TPNM} ;

Nep -> {DD} ;

Nes -> {OS,DS} ;

Neu -> {IN1,IN2,IN3,IN4,IN5,DN} ;

Nd -> {Ndabe,ND3,ND5}

DM ->{ND1,ND4,ONP,NOP1,NOP2,NOP4,NOP5,XQP,DSP,OSP,DDP1,DDP2,DSP3,ST
DM,RNOP1,RNOP2,RNOP3};

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 281-296

Tonal Errors of Japanese Students Learning Chinese: A Study of Disyllabic Words

Ke-Jia Chang*, Li-Mei Chen+ and Nien-Chen Lee+

Abstract

To foreigners, how to manage tone is the greatest challenge in learning Chinese. What causes foreign students to be unable to distinguish different tones is the phonological system of their native language. The accent in standard Japanese (Tokyo dialect) is distributed in the pitch change within each syllable, and the first syllable must be the opposite of the second in accent. The discrepancy between the tonal production of Japanese students learning Chinese and that of Chinese native speakers was investigated in this study. It is found that the two Japanese students in this study made the most frequent mistakes in reading Chinese disyllabic words when the first syllable was tone 2 or tone 3, and the tonal errors were mostly found in disyllabic words with tone combinations of 2-1, 2-4, and 3-4. We also found that in Group B (2-1, 2-2, 2-3, 2-4), whatever the original tones were, the two subjects always mispronounced them as 2-3. This is primarily attributed to the fact that, in Japanese, only one pitch peak is allowed in disyllabic compounds.

Keywords: Japanese Students Learning Chinese, Disyllabic Words, Tonal Errors

1. Introduction

One of the most distinct features of Chinese is tone, in which each syllable has its own fixed tone, including both high-low distinctions and rising-falling variations. The acoustic characteristics of tones are mainly determined by pitch value. Tones are relatively defined. This so called "relativity" is the stability of pitch within the pitch range of an individual speaker.

* Graduate Institute of Teaching Chinese as a Second Language, National Kaohsiung Normal University, Kaohsiung, Taiwan
+ Department of Foreign Languages and Literature, National Cheng Kung University, 1 University Rd., Tainan, Taiwan. Phone: (06)2757575 ext 52231
E-mail: leemay@mail.ncku.edu.tw
The author for correspondence is Li-Mei Chen.

[Received February 16, 2006; Revised August 11, 2006; Accepted September 5, 2006]

Some educators have suggested that Chinese learners could compensate for their tonal errors by practicing monosyllabic words. Yet, in actual classroom settings, it is found that the practice of tone combination is more important, especially that of disyllabic words. This is because both the learners and teachers often neglect the collaborative pattern of tones in spontaneous speech, such as the rules of tone sandhi and the patterns of tone combination. Language teaching should aim at a definite goal, and the teaching of tone combination ought to be focused on disyllabic words [Zhu 1997]. Zhu's argument is grounded in the following two reasons. First, almost all combination patterns of monosyllabic words in spontaneous speech are included in disyllabic words. Therefore, disyllabic words could be regarded as the foundation. Second, modern Chinese is mostly made up of disyllabic words. Practicing disyllabic words could solve most problems in tone combinations.

The changes of Chinese tone in connected speech pose a serious problem to Chinese learners. It is also found in classroom settings that Japanese students often stumble in communication because of their tonal errors. This paper studies the phenomenon of tonal errors in disyllabic words made by Japanese students learning Chinese, particularly in finding which tones these errors mostly occur in. It also investigates the negative transfer effect of the Japanese accent in learning Chinese tones by Japanese students, for the purpose of making certain contributions to Chinese pronunciation pedagogy.

2. Literature Review

2.1 The Phonetic Features of Chinese and Japanese

In Chinese, each syllable has its fixed tone. The high and low, falling and rising pitches depend on the vibration rate of the vocal cords (Figure 1). The constitution of Chinese tone is not determined only by pitch level, but also by transition patterns. There is a level tone, a rising tone, a falling tone, and a falling-rising tone which are caused by change in pitch. In addition to pitch, the intensity and duration of sound are also relevant to the make-up of the tone. Intensity indicates the weight or strength of a sound. For instance, the neutral tone in Chinese is related to sound intensity. The easiest and the most effective way to transcribe and record tones is the system of tone-letter proposed by [Chao 1968]. It classifies tone pitch into five degrees, and divides a perpendicular line into four parts to signify the particular location of the tone pitch on the scale. The low, mid-low, middle, mid-high, and high pitches are indicated by the numbers 1 to 5 respectively. The accurate tone-letter of each tone is represented by the high and low pitch, the rising and falling pitch, or the fluctuation of pitch. In a Chinese disyllabic phrase, the tones of the first and the second words are compromised for the sake of being euphonious [Wu 1992]. It is natural to make the pitch in the second syllable lower than that in the first. Take a disyllabic word with two rising pitches for example, the

second rising pitch turns into low-rising (Figure 2). In a disyllabic word with two falling pitches, both syllables are lower due to the mutual influence of these two falling pitches (Figure 3).

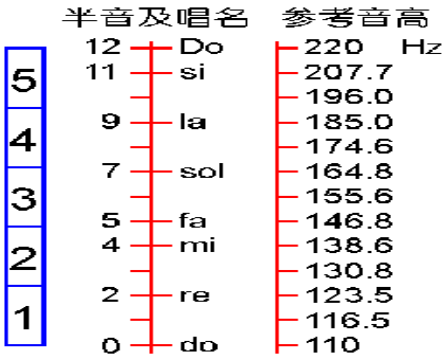
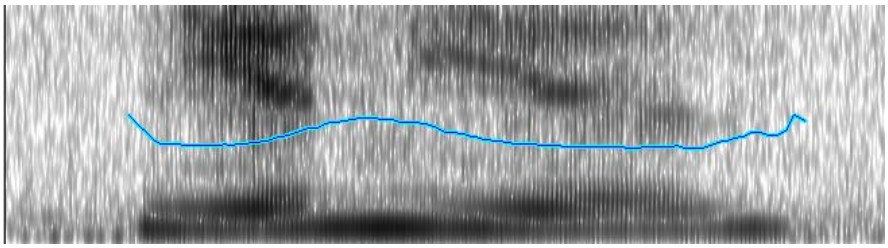
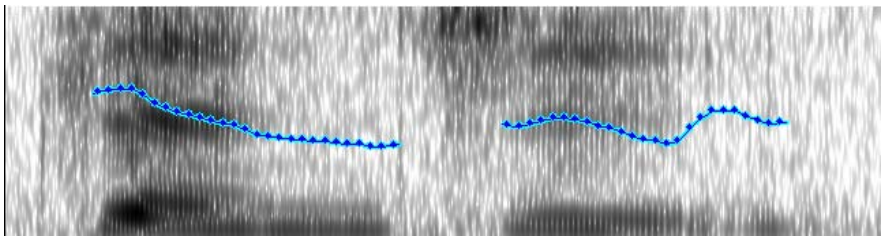


Figure 1. Frequency of calibration



Nian Nian

Figure 2. Word of tone 2-2 (年年, year-year) pronounced by the Chinese native speaker. (The blue line signifies tone.)



Jheng Jh

Figure 3. Word of tone 4-4 (政治, politics) pronounced by the Chinese native speaker.

In Japanese phonetic features, accent bears the closest relationship to Chinese tone. There are two types of accents in the languages of the world [Hiroshi 2003]. One is "stress accent", which uses the intensity of sounds to differentiate various lexical items. The other is "pitch accent", which uses the pitch of sounds to distinguish one word from another. Japanese

is classified as pitch accent language (termed as "accent" in the following sections). According to several researchers [Wang 1997; Jun 2001], the Japanese accent can be classified into two types--flat and non-flat. Wang also mentioned that some moras in Japanese must be pronounced with high pitch, and some with low pitch. "Mora" in Japanese means the duration of a sound. The accent in Japanese displays in the mora instead of in the syllable. For example, the word "shinbun" (news) has two syllables but four moras.

The difference between flat and non-flat type lies in the existence of the accent. The accent means there is a transition from high to low pitch in a word. The flat type does not have the accent whereas the non-flat type does. Falling type can also be classified into three patterns—H-L, L-H-L, and L-H. In the Compact Japanese-Chinese Dictionary [Liu *et al*. 2002], pitch change is illustrated by ◎, ①, ②, ③, ④, ⑤ at the end of the word. ◎ means that the first mora is pronounced with low pitch and the remaining moras are produced with high pitch, which may spread to the subsequent auxiliary. This is the flat type, such as hashi (chopsticks), and tomodachi (friend). ① means the first mora is pronounced with high pitch whereas the subsequent with low pitch. This tone falls into the H-L pattern, such as neko (cat). ② means that the first mora is pronounced with low pitch, the second with high pitch, and the subsequent with low pitch, including the following auxiliary. The words composed of two moras in this tone belong to the L-H pattern, while those composed of three or more moras belong to the L-H-L pattern, such as kawa (river) and nomimono (beverage). ③ means the first mora is pronounced with low pitch, the second and the third with high pitch, and the subsequent as well as the following auxiliary with low pitch. Words composed of three moras of this tone fall into the L-H pattern, while those composed of more than four moras belongs to the L-H-L pattern, such as otoko (man) and mizuumi (lake). ④ means that the first mora is pronounced with low pitch, the second to fourth with high pitch, and the subsequent with low pitch. Words composed of four moras in this tone belong to the L-H pattern, while those composed of five or more moras follow the L-H-L pattern, such as otouto (junior brother) and watashibune (ferry boat). ⑤ means the first mora is pronounced with low pitch, the second to the fifth with high pitch, and the subsequent mora with low pitch. In this tone, words with five moras belong to the L-H pattern, while those composed of six or more moras use the L-H-L pattern, such as oshougats (New Year) and tansangasu (carbon dioxide).

According to the patterns stated above, the accent in standard Japanese (Tokyo dialect) has the following characteristics. First, there can only be one part with high pitch in a word (with one mora or several consecutive moras). Second, the pitches of the first and the second moras must differ. If the first mora is pronounced with high pitch, the second one must be with low pitch. In the same way, if the first is with low pitch, the second must be with high pitch. Third, the pitch change in the first and third tones in Chinese does not occur in the Japanese accent.

The Japanese accent and Chinese tone seem to be represented by pitch change. In Japanese, the accent is represented in the pitch change of each mora within a word. The basic component is a mora. However, in Chinese, tone is displayed in the pitch change within each syllable. The basic unit is a morpheme. There are four basic tones in Chinese (except for the neutral tone)—the high-level tone, mid-rising tone, low-falling tone, and high-falling tone. In terms of tone values, they are marked as 55, 35, 214, and 51 respectively (Table 1).

Table 1. The diacritics in the system of tone-letter designated by [Chao 1968]

tone types	Yinping (High-level)	Yangping (mid-rising)	Shangsheng (falling-rising)	Qusheng (high-falling)
tone values	55	35	214	51
examples	**mā**	**má**	**mǎ**	**mà**
duration	mid-short	mid-long	longest	shortest

2.2 The Tonal Errors of Japanese Students Learning Chinese

There are three common errors made by Japanese students in learning Chinese [Zhu 1994]—flat tone, mispronunciation of multi-syllabic words, and stress of the neutral tone. Many Japanese students of Chinese pronounce disyllabic words in Chinese with rising-falling tones, regardless of their original tones, such as changing "chun1fong1" (spring breeze) to "chun2fong1" (pure breeze), and changing "fang1bian4" (convenient) to "fang2biang4" (room convenient). The cause of this mispronunciation is related to the "euphonic change" in Japanese. Whatever the original pitch pattern is, when two words are combined into one lexical item, only the L-H-L pattern is allowed. For example, the original pitch of "waseda" belongs to the H-L pattern while that of "daigaku" (university) the L-H pattern. When these two words are combined, the pitch of "wasedadaigaku" (Waseda University) turns into the L-H-L pattern. This is because in Japanese, there cannot be two pitch changes in one word, which means that only one pitch peak is allowed in Japanese compounds.

It is very difficult for Japanese students to distinguish tone 3 from tone 4, tone 2 from tone 3, and tone 2 from tone 4 in Chinese [He 1997]. They easily mistake tone 3 for tone 2.

3. Methodology

3.1 Subjects

The subjects in this study were two Japanese students with a basic-intermediate level in Chinese. Both of them were from the Chinese Learning Center in National Sun Yat-sen University and had studied Chinese for three to six months. Subject X had better Chinese ability than subject Y. There was another subject Z, whose native language is Chinese, serving as the control group in this study. All subjects were required to read out the disyllabic words listed in the word chart in the same manner.

3.2 Procedure

This study is divided into three parts. The first part is to make real-life interviews so as to collect natural data to supplement the word chart. The second is to ask the subjects to read out the disyllabic words in the word chart. In order to maintain the objectivity of this research, the word chart is divided into two lists, A and B, in which the contents are completely the same with only different arrangement of the order. The design of the word chart primarily follows that of [Zhu 1997].

3.3 Design of Word Chart

There are four tones in Chinese. If all four tones are arranged into disyllabic words, sixteen combination pairs are retrieved. Including the neutral tone, there are twenty possible combination pairs. In this study, these twenty pairs are divided into five groups--A, B, C, D, and E. The number 1, 2, 3, 4, 5 represent the high pitch, rising pitch, falling-rising pitch, falling pitch, and neutral tone, respectively, as illustrated below.

A：1-1、1-2、1-3、1-4

B：2-1、2-2、2-3、2-4

C：3-1、3-2、3-3、3-4

D：4-1、4-2、4-3、4-4

E：1-5、2-5、3-5、4-5

To avoid expectation of a pattern from the subjects, each group in the word chart has been rearranged. The word chart has been supplemented with Chinese phonetic symbols (bpmf) and all the disyllabic words listed come from basic vocabulary. Before the recording, the subjects were familiarized with the demo word chart with no time limit, and were not

informed of the correct pronunciation. During the formal recording, if the subjects made any mistakes, they were allowed to self-correct with assistance from others prohibited. The third part of this experiment was to ask Chinese native speakers to take auditory tests and pick up the tonal errors of subject X and subject Y. The tonal errors were analyzed with the phonetic analysis software PRAAT.

3.4 Methods of Analysis

First, we identified the tonal errors made by subject X and Y in pronouncing these five groups of sounds (A, B, C, D, E), and judged which subject had the most errors. Then, we investigated the ratio of tonal errors of these two Japanese subjects in each group of sounds to draw a comparison to the tone production of Chinese native speakers.

4. Results and Discussion

4.1 Ratio of Tonal Errors

From Table 2 we can see that the tonal errors of Subjects X and Y are mostly concentrated in the sounds in Groups B and C. In other words, they are mostly compounds consisting of a first syllable that is tone 2 or tone 3.

Table 2. Ratio of tonal errors of subjects X and Y

Second Syllable / First syllable	1		2		3		4		5(E)	
Subjects	X	Y	X	Y	X	Y	X	Y	X	Y
1(A)	0.333	0.333	0.2	0	0.3	0.3	0.149	0.285	0.2	0.2
2(B)	**0.4**	**0.8**	0.25	**0.75**	0	**0.5**	**0.4**	**1**	**0.667**	0.333
3(C)	**0.375**	0.375	0.286	**0.625**	0	**0.5**	**0.4**	**0.8**	0.25	**0.75**
4(D)	0.333	0	0.333	0	0.167	0	0.2	**0.4**	0	0

Since the two subjects do not have the exactly same Chinese background, the ratio of tonal errors is compared with their individual average number. The ratio of tonal errors of subject X is 0.336 on average whereas that of subject Y is 0.4. Therefore, for subject X the ratio of tonal errors of more than 0.33 is high, while for subject Y 0.4. In the sounds of Group C (3-3), the ratio of tonal errors of subject Y appears high, which may imply that the subject has not been fully acquainted with the rules of tone sandhi.

Table 3. Ranking of ratio of errors in subjects X and Y

Ranking of ratio of errors in subject X	Ranking of ratio of errors in subject Y
1. 2-5	1. 2-4
2. 2-4、3-4、2-1	2. 3-4、2-1
	3. 3-5、2-2
	4. 3-2
	5. 2-3、3-3

Although it is hard to see the similarity in distribution from the ranking of the two subjects' ratio of tonal errors (Table 3), two interesting phenomena are found. First, by comparing their high ratio of tonal errors, it is shown that there is overlap in 2-1, 2-4, and 3-4. Second, the ratio of errors of 2-3 and 3-3 are the same in both subjects. The similarity of the results obtained from these two groups is mainly contributed to the fact that the tone sandhi of 3-3 is 2-3. However, the ratio of errors of subject Y in pronouncing 3-3 reaches as high as 0.5, which indicates that he has not yet fully managed the tone sandhi rules. What follow in the next section are individual tonal errors in each tone combination group.

4.2 Tonal Errors in Each Group

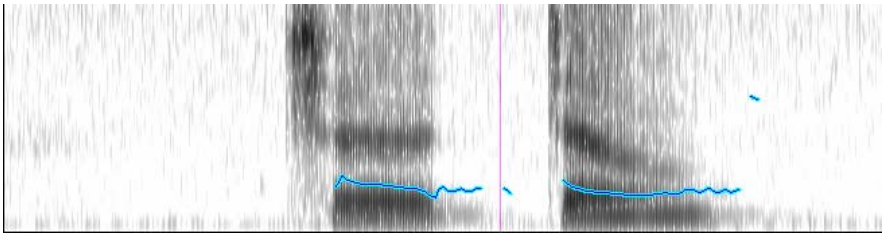
Table 4. Error patterns of Group A in Subject X

Standard	1-1	1-2	1-3	1-4
Mispronunciation	1-4	2-2	2-3	4-4

Table 5. Error patterns of Group A in Subject Y

Standard	1-1	1-2	1-3	1-4
Mispronunciation	4-2	---	1-3/(1-2)	1-1 1-1

In Group A, we discover the error pattern of "rising falling tone" in the corpus of subject X, in which 1-2 and 1-3 are mispronounced as 2-2 and 2-3, respectively (Table 4). It is also found both subjects often mispronounce the first tone as the second or the fourth, as was indicated as a common error made by Japanese students [Chao 2003]. Subject Y always mispronounces 1-4 as 1-1 (Table 5). It is assumed that the subject is unable to articulate the fourth tone while the pitch of the first syllable remains as the first tone (Figures 4 and 5).



Chejhan (Subject Y) 1-4 ⟶ 1-1(mispronounced)

Figure 4. Mispronunciation of "chejhan" (車站, train station) by subject Y

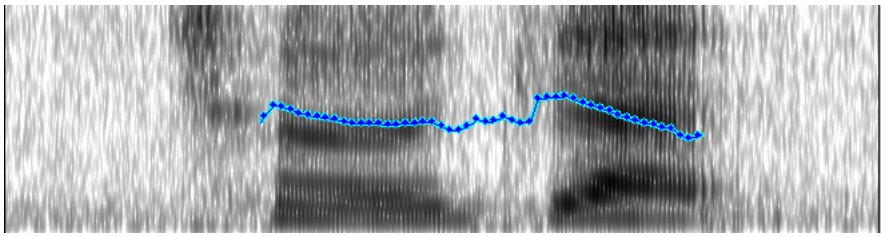


Figure 5. Pronunciation of "chejhan" (車站, train station) by the Chinese native speaker

Table 6. Error patterns in Group B in Subject X

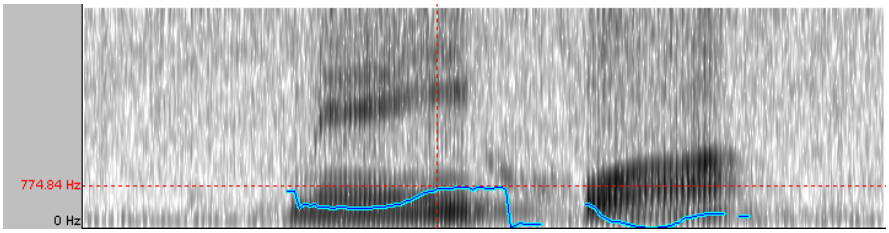
Standard	2-1	2-2	2-3	2-4
Mispronunciation	1-1 1-1(2-2)	2-3 3-1(2-1)	---	1-4 1-4

Table 7. Error patterns in Group B in Subject Y

Standard	2-1	2-2	2-3	2-4
Mispronunciation	1-1 2-3	1-1 1-1/(2-2) 1-2/(2-2) 1-2 1-2 2-1	1-2/(1-3) 1-3	1-4 1-4 2-3/(2-4) 1-4

In the sounds of Group B, both subject X and Y mispronounce tone 2 as tone 1 (Tables 6 and 7), but with lower pitch than that of the native speaker. It is also found that the tone of the second syllable is mispronounced as well. In the sounds of Group B, there are four sounds of which the first syllable is correctly uttered but the second is mispronounced (if the rectified productions are not taken into account). For example, 2-2 is mispronounced as 2-3 by subject X and 2-1 as 2-3 by subject Y, 2-2 as 2-1 and 2-4 as 2-3. According to the findings above, both subjects make frequent errors in mispronouncing the sounds in Group B as 2-3. This is because in the Japanese accent only one pitch peak is allowed in a word, and after a high pitch

there will only come a low pitch. It is rather difficult for Japanese students to maintain high pitch after the end of the second tone in 2-1 (Figures 6 and 7).



Meihua (Subject Y) 2-1 ⟶ 2-3 (Mispronunciation)

Figure 6. Mispronunciation of "meihua" (梅花, rosette) by subject Y

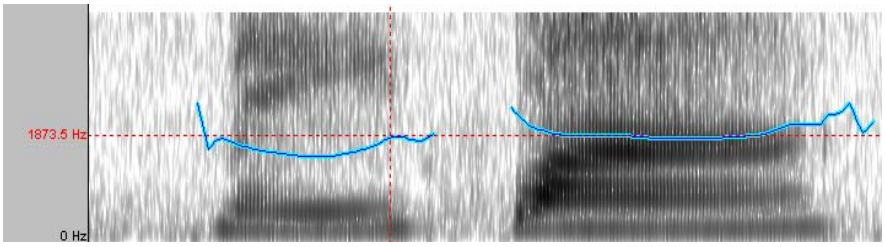


Figure 7. Pronunciation of "meihua" (梅花, rosette) by the Chinese native speaker

Table 8. Error patterns in Group C in Subject X

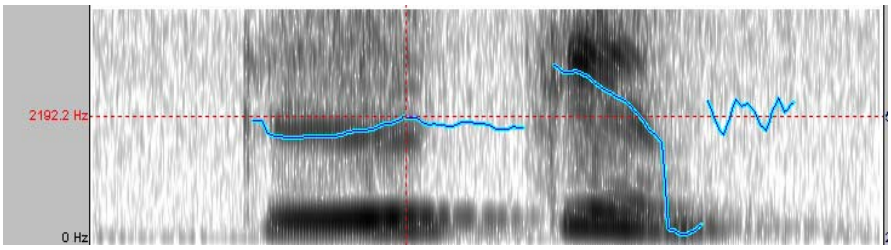
Standard	3-1	3-2	3-3	3-4
Mispronunciation	2-1 4-1 1-1/(2-1)	3-1 3-3/(2-3)	----	2-1 2-4

Table 9. Error patterns in Group C in Subject Y

Standard	3-1	3-2	3-3	3-4
Mispronunciation	4-1 1-1 1-1	2-3 1-2 1-1/1-2 2-1 2-2/1-1	1-3 1-3 1-3	2-1 2-4/(2-3)/ 1-3 2-1 1-4

In Group C, tone 3 of the first word is often mispronounced as tone 1 or tone 2 (Tables 8 and 9). Judging from the ratio of zero error of subject Y in pronouncing 3-3, the subject can completely manage the tone sandhi rules for the third tone. Similarly, it is found that subject X also has a ratio of errors of zero in pronouncing 2-3. As for subject Y, this subject's ratios of tonal errors are both 0.5 in pronouncing 3-3 and 2-3, which indicates the subject has not yet

been fully acquainted with the tone sandhi rules of the third tone. From Figures 8 and 9, one can see that the Chinese native speaker has a longer duration of 21 in pronouncing 214 of the sound "jhu" (self), whereas the duration of 14 is rather short. On the contrary, Japanese students have longer duration of 14 in pronouncing 214 whereas that of 21 is very short. As a result, it sounds like tone 2.



Jhujian (Subject X) 3-4 ⟶ 2-4(Mispronunciation)

Figure 8. Mispronunciation of "jhujian" (主見, self opinion) by subject X

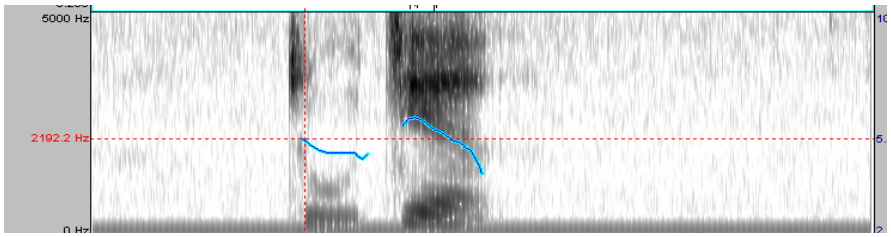


Figure 9. Pronunciation of "jhujian" (主見, self opinion) by the Chinese native speaker

Table 10. Error patterns in Group D in Subject X

Standard	4-1	4-2	4-3	4-4
Mispronunciation	1-1	----	1-3	1-4

Table 11. Error patterns in Group D in Subject Y

Standard	4-1	4-2	4-3	4-4
Mispronunciation	---	2-1	---	1-4 4-1

Group D has the lowest ratio of tonal errors, many of which are zero (Tables 10 and 11). Subject X mispronounces tone 4 of the first word as tone 1 most frequently, in keeping with what Zhu had proposed that this mispronunciation bears the features of rising-falling pattern. Since the tone value of the neutral tone is determined by the tone of the preceding syllable, it will be discussed in another section.

4.3 Ratio of Tonal Errors in Group E

Table 12. Error patterns in Group E in Subject X

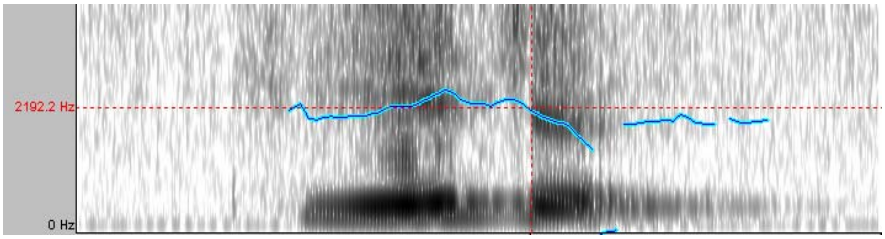
Standard	1-5	2-5	3-5	4-5
Mispronunciation	2-5	2-1	2-5	----

Table 13. Error patterns in Group E in Subject Y

Standard	1-5	2-5	3-5	4-5
Mispronunciation	2-5	1-5	2-5	----
		3-5	1-5	
			1-5	

Zhu has mentioned that neutral tone is often dealt with as tone sandhi in phonetic analysis as its tone values are determined by the tone in the preceding syllable. After tone 1 and tone 2, the pitch value of the neutral tone is 31; after tone 3, the pitch value is 4; after tone 4, the pitch value is 1. The duration of neutral tone is, in general, shorter. Although the pitch of the neutral tone is usually light and short, it is not invariable. The pitch of neutral tone is always changed according to what the end of preceding syllable is. A common error of subjects X and Y is the mispronunciation of 1-5 as 2-5 (Tables 12 and 13). In the following section, the researchers will investigate the discrepancy between Chinese native speakers and Japanese students of Chinese in pronouncing the neutral tone in disyllabic words.

When the first syllable is tone 1, it is found that the Japanese student has a greater degree of descent than that of the Chinese native speaker (Figures 10 and 11).



Jiejhe (Subject A) 1-5 ⟶ 2-5(Mispronunciation)

Figure 10. Mispronunciation of "jiejhe" (接著, next) by subject X

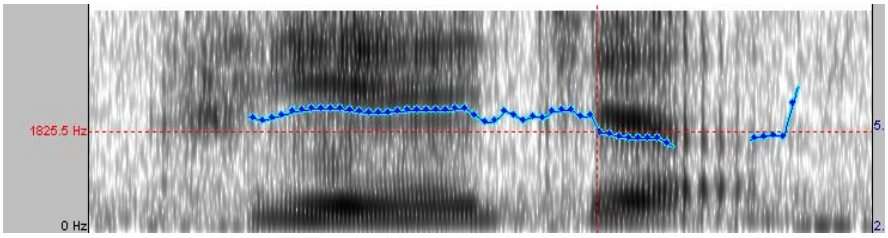
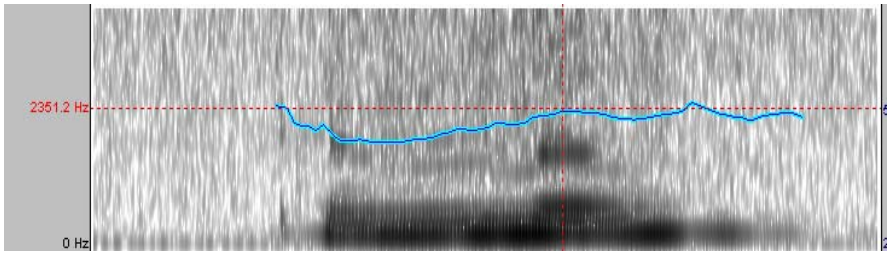


Figure 11. Pronunciation of "jiejhe" (接著, next) by the Chinese native speaker

In “men” of “Renmen”, it is found that the Chinese native speaker pronounces the word with short duration whereas the Japanese student prolongs the word with the neutral tone (Figures 12 and 13).



Renmen (Subject X) 2-5 ⟶ 2-1(Mispronunciation)

Figure 12.Mispronunciation of “renmen” (人們, people) by subject X

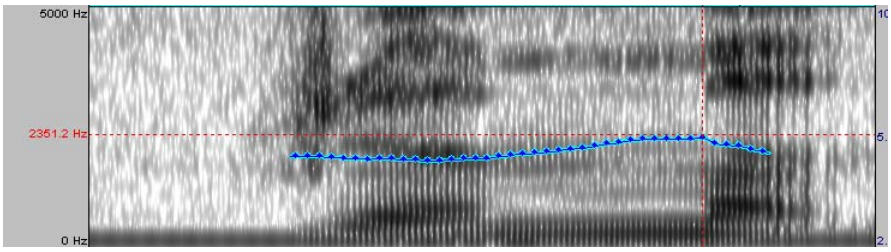
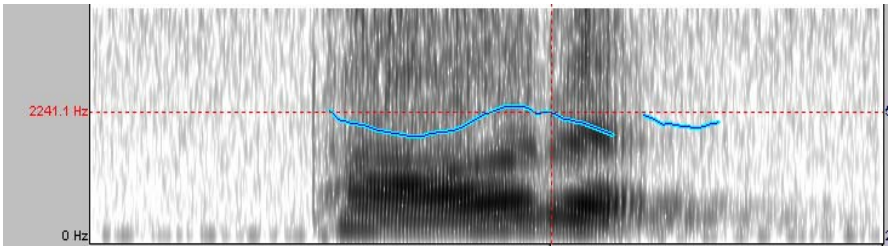


Figure 13. Pronunciation of “renmen” (人們, people) by the Chinese native speaker

While the Chinese native speaker is pronouncing “paole”, the native speaker makes a slight rise in tone at the end of the second syllable. As for the Japanese student, the first sound is mispronounced as tone 2. Therefore, the tone shape is displayed with a descending curve (Figures 14 and 15).



Paole (Subject X) 3-5 ⟶ 2-5

Figure 14. Mispronunciation of “paole” (跑了, have run) by subject X

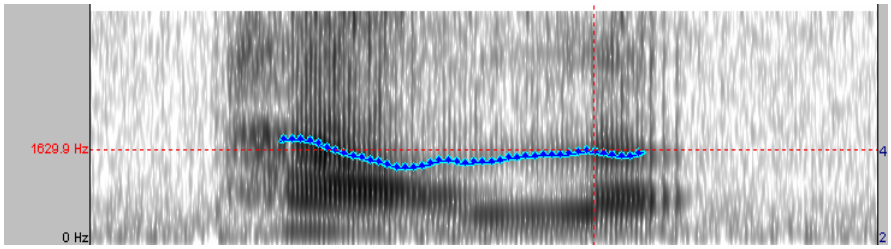
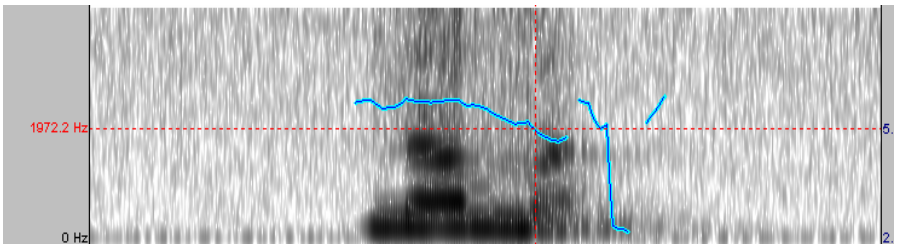


Figure 15. Pronunciation of "paole" (跑了, have run) by the Chinese native speaker

In the following example, it was also found that the Japanese student pronounced the second syllable with excess high pitch (Figures 16 and 17).



Na me (Subject X) 4-5 ⟶ 1-5(Mispronunciation)

Figure 16. Mispronunciation of "na me" (那麼, so) by subject X

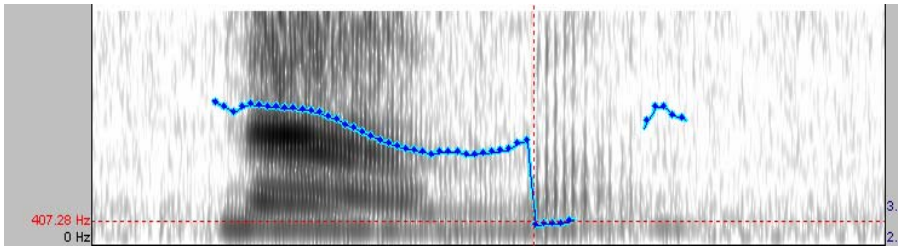


Figure 17. Pronunciation of "na me" (那麼, so) by the Chinese speaker

5. Conclusion

This study focuses only on disyllabic words. In Japanese, there are tone variations and euphonic changes in words of more than two syllables. A rising-falling pattern was proposed [Zhu 1994] based on the pitch change of Japanese students and it was argued that regardless of the original pitch in Japanese, new compounds are always of Low-High-Low tones. The tonal errors of the two Japanese students in this current study are also similar to this pattern because of the negative transfer from the students' mother tongue.

The tonal errors made by both subjects X and Y concentrated primarily in Groups B and C. The mispronunciation is mostly on words with first syllable of tone 2 or tone 3. Most of the errors are in 2-1, 2-4, and 3-4 tone combinations. Moreover, the ratio of tonal errors of 2-3 and

3-3 are completely identical in these two subjects. From the tonal errors in Group C, the researchers found that the two students are confused with tone 3, tone 2, and tone 1.

In teaching Chinese tones, it is suggested that teachers could start from the pronunciation and listening comprehension of disyllabic words, instead of merely concentrating on drilling students using monosyllabic words. Teachers could first familiarize the learners with tone combinations by practicing tone variations in disyllabic words. In class, visual demonstrations such as graphs, gestures, and animated films could be utilized to help students understand tone variations. Generally speaking, whatever Chinese proficiency the students might have, they still cannot precisely distinguish different tones. Therefore, teachers should emphasize more on practicing contrastive tones to get students acquainted with the tone combinations in disyllabic words.

The authors only discuss the tonal errors of two subjects in this study. It is suggested that, in future study of related issues, more Japanese and Chinese subjects should be included to make the experimental results more representative. Moreover, tests on perceptual distinction could be added in further studies to obtain a more complete picture of the acquisition of Chinese tones.

Since the functions of Chinese tone and Japanese pitch accent differ, by means of contrastive analysis one can help teachers pay special attention to those tones Chinese learners frequently get confused, so as to make the learners fully acquainted with correct tone production in various tone combinations.

References

Chao, L.- J., "Special Phonetic Instruction in Chinese to Japanese Students," *Journal of Yunnan Normal University*, 1(3), 2003, pp. 66-67.

Chao, Y. R., *A Grammar of Spoken Chinese*, Berkeley: University of California Press, 1968.

He, P., "Studies in Basic Phonetic Instruction in Chinese to Japanese Students," *Language Teaching and Linguistic Studies*, 3, 1997, pp. 49-50.

Hiroshi, W., "The Theory of Japanese Stress and Phonetic Categorization," *Journal of Wu Feng Institute of Technology,* 11, 2003, pp. 283-286.

Jun, S., "Problems and Solutions of Japanese Learning Chinese Tones," *Proceedings of the International Symposium in Teaching Chinese to Japanese Students, 172.* Peking: Chinese Social Science, 2001.

Liu, W.- S., J.-S. Ma, Y.- H. Tzeng, S.- K. Li, J.-J. Huang, and S.- S. Wang, *Compact Japanese-Chinese Dictionary,* Taipei: Tashin, 2002.

Wu, Z.- J., *Introduction to the Phonetics of Modern Chinese,* Peiking: Chinese Language Teaching, 1992.

Wang, S.- Y., "Japanese Tone and Pronunciation," *Foreign Languages in Fujian*, 2, 1992, pp. 24-26.

Zhu, C., "Contrastive Experiments on the Suprasegmental Features of Chinese and Japanese," *Journal of East China University*, 1, 1994, pp. 85-86.

Zhu, C., *The Strategies of Foreign Students for the Phonetic Learning in Chinese.* Peking: Language, 1997.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 3, September 2006, pp. 297-314

Performance Analysis and Visualization of Machine Translation Evaluation

Jianmin Yao[*+], Yunqian Qu[+], Qiang Lv[+], Qiaoming Zhu[+], and Jing Zhang[**]

Abstract

Automatic translation evaluation is popular in development of MT systems, but further research is necessary for better evaluation methods and selection of an appropriate evaluation suite. This paper is an attempt for an in-depth analysis of the performance of MT evaluation methods. Difficulty, discriminability and reliability characteristics are proposed and tested in experiments. Visualization of the evaluation scores, which is more intuitional, is proposed to see the translation quality and is shown as a natural way to assemble different evaluation methods.

Keywords: Machine Translation, Performance, Analysis, Visualization, Clustering, Natural Language Processing

1. Introduction

Machine translation (MT) evaluation activities have accompanied MT research and system development. The ALPAC report [ALPAC 1966], which has greatly influenced machine translation research activities, is the first historical MT evaluation activity. With new developments in natural language processing technology coming in the 1990s, the black-box evaluation has been instantiated by the methodology of DARPA [Doyon *et al*. 1998], which measures fluency, accuracy, and informativeness on a 5-point scale. The ISLE Project takes an approach that focuses on how an MT system serves the follow-on human processing rather than on what it is unlikely to do well [ISLE 2000].

Since manual evaluation is labor-intensive and time-consuming, many researchers are making efforts towards reliable automatic MT evaluation methods. A problem is that the methods cannot be characterized by precision and recall as in other natural language

[*] Southeast University, Nanjing, China 210096 Tel: +86-512-68880263
E-mail: jyao@suda.edu.cn
[+] School of Computer Science and Technology, Soochow University, Suzhou, China, 2150006
[**] South China University of Technology, Guangzhou 510641, China

[Received March 21, 2006; Revised October 11, 2006; Accepted October 16, 2006]

processing activities such as POS tagging or phrase identification. A new quality system is necessary.

This paper aims for performance analysis and better illustration of machine translation evaluation, which can help developers know about the improvement in the quality of their system, and help users easily distinguish between MT systems. Section 2 reviews related research in the MT field and its evaluation. Section 3 studies the metrics and experiments for comparison of MT evaluation methods. Section 4 proposes an algorithm for visualizing the MT system quality, and draws a dendrogram for the systems by clustering. A conclusion is given in the last section.

2. Related Work

MT evaluation had not been a very powerful aid in machine translation research until automatic evaluation methods were broadly studied. Now, different heuristics are employed for automatic MT evaluation. This section gives a brief review of the main automatic MT evaluation methods and studies on the performance of these methods.

2.1 Automatic Evaluation Methods

Some automatic methods focus on specific syntactic features for translation evaluation. [Jones and Galliers 1993] utilizes linguistic information such as the balance of parse trees, N-grams, semantic co-occurrence, and other information as indicators of translation quality. A balanced tree was a negative indicator of Englishness, probably because English is right-branching. Other factors are also utilized in translation evaluation for their indication of the language quality. [Brew and Thompson. 1994], whose criteria involve word frequency, POS tagging distribution and other text features, compares human rankings and automatic measures to decide the translation quality. These linguistic features are extracted as a reflection of the overall translation quality.

Another type of evaluation method involves comparison of the translation result with human translations. [Keiji *et al*. 2001] evaluates the translation output by measuring the similarity between the translation output and translation answer candidates from a parallel corpus. [Yasuhiro *et al*. 2001] uses multiple edit distances to automatically rank machine translation output by translation examples. While the IBM BLEU method [Papineni *et al.* 2001] and the NIST MT evaluation [NIST 2002] compare MT output with expert reference translations in terms of the statistics of word N-grams. [Melamed *et al.* 2003] adopts the maximum matching size of the translation and the reference as the similarity measure for the score. [Niben and Och 2000] scores a sentence on the basis of scores of translations in a database with the smallest edit distance. [Yokoyama *et al.* 1999] proposes a two-way MT based evaluation method, which compares output Japanese sentences with the original

Japanese sentence for word identification, the correctness of the modification, the syntactic dependency and the parataxis.

Another path of MT evaluation is based on test suites. A weighted average of the scores for separate grammatical points is taken as the score of the system. The typological test covers vocabulary size, lexical capacity, phrase, syntactic correctness, etc. [Yu 1993] designs a test suite consisting of sentences with various test points. [Guessoum and Zantout 2001] proposes a semi-automatic evaluation method of the grammatical coverage machine translation systems via a database of unfolded grammatical structures. [Koh *et al.* 2001] describes their test suite constructed on the basis of fine-grained classification of linguistic phenomena.

2.2 Performance of an Automatic Evaluation Method

The ISLE has made some efforts to develop a specification of performance for the MT evaluation methods [ISLE 2000]. A list of the desiderata demands that at least the measure: 1) must be easy to define, clear, and intuitive; 2) must correlate well with human judgments under all conditions, genres, domains, etc.; 3) must be 'tight', exhibiting as little variance as possible across evaluators, or for equivalent inputs; 4) must be cheap to prepare; 5) must be cheap to apply; 7) should be automated, if possible. These criteria give a broad coverage of the characteristics of the evaluation methods, but further work is needed to measure them in a consistent and objective way.

[Popescu-Belis 1999] argues that the MT evaluation metrics should have its upper limit, lower limit, and should be monotonic in quality measure. The above measures are qualitative attributives of MT evaluation methods. If it can further be automated, it will help the researchers find a much easier and consistent way to compare different systems.

Only recently, researchers began quantitative studies. Some recent works include [Forner and White 2001] on the correlation between intelligibility and fidelity and noun compound translation. [Papineni *et al.* 2001] and [Melamed *et al.* 2003] study the correlation between human scoring and automatic evaluation results. After DARPA took the BLEU method as the evaluation method for MT systems, the correlation between human and machine translation evaluation has become a standard criterion of MT quality scoring, though many researchers are arguing against its efficacy.

On the whole, methodological study of automatic evaluation methods has just started and needs to be further deepened. This paper is an attempt to refine the correlation measures and justify their usage in machine translation evaluation. The following section aims for a proposal of some criteria of the performance of MT evaluation measures, which will give linguists a better understanding of the MT evaluation task and its results.

3. MT Evaluation Performance Analysis

Up to now, the analysis of MT evaluation methods has remained a preliminary comparison of human and automatic scores. Further study is important to propose better evaluation measures and better understanding of the automatic evaluation results. This paper is an endeavor to provide more details of MT evaluation methods. A list of quantitative measures on basis of education measurement theory [Wang 2001] is proposed in section 3.1, and experimental study of the measures is made in section 3.2.

3.1 MT Evaluation Performance Metrics

3.1.1 Consistency and Reliability

Reliability is the most important issue in MT evaluation. Correlation is often utilized for description of the consistency between different score results as by various MT evaluation methods or test suites, as follows:

$$r_{tt} = \frac{\sum X_a X_b - (\sum X_a)(\sum X_b)/n}{\sqrt{\sum X_a^2 - (\sum X_a)^2/n}\sqrt{\sum X_b^2 - (\sum X_b)^2/n}}, \quad (1)$$

where X_a and X_b refer to scores of the two MT evaluation results; n is the number of test questions in the test suite; r_{tt} is the consistency between the two test results. If the scores are rank-based, reliability can be calculated by Spearman rank correlation as

$$r_{tt} = 1 - \frac{6\sum D^2}{n(n^2-1)}, \quad (2)$$

where D is the difference between ranks of the same test by different evaluators; n is the sample size.

The correlation coefficient between the automatic results and the human results shows the reliability of the automatic evaluation method. On the other hand, if the correlation is between two automatic results, it shows consistency between the two methods, thus, also showing whether they can compensate for each other.

3.1.2 Discriminability

The discriminability of an MT evaluation method reflects the ability to distinguish between minor differences in translation qualities. For a test with higher discriminability, a better system should be scored higher, and vice-versa. The MT evaluation result should be fine-grained so that even small changes in the translation quality could be correctly shown. The discriminability of a test can be calculated on the basis of the MT evaluation result, as

follows:

$$D = (X_H - X_L)/(H - L) . \quad (3)$$

In the equation, X_H / X_L is the score for the best/worst system; H / L is highest/lowest possible score of the test.

3.1.3 Difficulty

The difficulty refers to the degree of the difficulty of the test, which has a great influence on the test result. The difficulty of the test changes the distribution, discriminability, and dispersion of the scores. For example, if the test is so difficult that none of the systems outputs the right answer, one cannot distinguish between systems via the MT evaluation result. This is also the case if the test is too easy. The difficulty of the test questions can be calculated as

$$P = (\bar{X} - L)/(H - L) . \quad (4)$$

In the equation, $\bar{X}$ is the average score of the systems, while H/L is the highest/lowest possible score for the test. The difficulty of the test question is closely interrelated with the discriminability, efficacy, and other characteristics of the evaluation. According to education measurement theory, a difficulty of around 0.5 is helpful for discriminating the systems to be scored [Wang 2001].

In the section above, a proposal of performance metrics for MT evaluation measures and the proposal's test suite has been given. These metrics help in analyzing the efficacy of the evaluation methods. The next section gives some experimental examples of the evaluation performance, which verifies the metrics mentioned above.

3.2 Experiments on MT Evaluation Performance

3.2.1 Test of Consistency, Discriminability and Difficulty

Since the MT evaluation performance metrics proposed in section 3.1 are language-independent, they can be applied to evaluation results in any language. The open source of human evaluation results in [Darwin 2001] on eight English-to-Japanese MT systems is taken for analysis in this section. The authors of this paper do research on the open source evaluation results for two reasons: it is available to any researcher, and thus is easier to duplicate the experiment and analysis; also, the open source data is appropriate in data size and reliability and saves time for more manual work. In the experiment in [Darwin 2001], two evaluators score 8 systems on a 5-point scale showing intelligibility and accuracy. The experimental setup and details are listed in the appendix following this paper. Based on the measures proposed in the last section, this paper's authors make an analysis of the

characteristics of the MT evaluation results.

The first experiment is to test the consistency between MT evaluation results from different measures (accuracy and intelligibility), different evaluators, and different test suites. According to equation (1) and (2), based on the data in Table A1 and A2 in the appendix, one gets the correlation coefficients in Table 1, which shows the correlation coefficients for the MT evaluation results.

In Table 1, rows 1 and 2 show a consistency between MT evaluation results by metrics of intelligibility and accuracy. Rows 3 to 5 show consistency between two human evaluators A and B. Rows 6 to 8 show consistency between MT evaluation results by the same evaluator A on different parts of the 300 hundred sentences.

Table1. The correlation coefficients for the MT evaluation results achieved from different evaluation measures of intelligibility and accuracy), different evaluators (named as A and B) and various test suites (3 parts of 300 sentences).

Item1	Item2	Other conditions	Correlation option	Correlation
Intelligibility	Accuracy	Overall average scores	Pearson	0.998
Intelligibility	Accuracy	Overall average scores	Spearman	1.000
Evaluator A	Evaluator B	Intelligibility for all 300 sentences	Pearson	0.991
Evaluator A	Evaluator B	Accuracy for all 300 sentences	Pearson	0.998
Evaluator A	Evaluator B	Accuracy for all 300 sentences	Spearman	0.994
Sent#1-100	Sent#101-200	Intelligibility evaluator A	Pearson	0.964
Sent#1-100	Sent#201-300	Intelligibility evaluator A	Pearson	0.968
Sent#101-200	Sent#201-300	Intelligibility evaluator A	Pearson	0.945

From the definition in section 3.1, one knows that correlation between different human evaluation results is an upper bound of automatic MT evaluation performance. Correlation with a human evaluation also reflects the reliability of the automatic evaluation result. As seen in Table 1, all correlation coefficients are higher than 0.9, which is a strong hint of consistency. First, the correlation coefficient between intelligibility and accuracy are 0.998 and 1.000, respectively. This reminds researchers that the two metrics have quite similar scores, and a researcher may just measure one and know the other by regression analysis. Second, the coefficient is also high for correlation between different evaluators and different parts of the test suite, which shows that scores from both evaluators and from different sentences agree with each other on the whole. This is also the case for automatic measures. From previous study, one knows that some automatic evaluation methods are highly correlated with human evaluation, for example, a correlation of around 0.99 for BLEU and NIST [NIST 2002]. GTM (General Text Matching) claims a 0.8 level which is better than BLEU on the same test suite

[Melamed *et al*. 2003]. The difference between [Melamed *et al.* 2003] and [NIST 2003] gives researchers a strong signal that consistency is a key factor, but not the only one, in MT evaluation performance.

Another key issue seen from Table 1 is that rows 6 to 8 have a lower correlation coefficient than the rows above. It reminds the researchers that different metrics, such as intelligibility and accuracy, different evaluator A and B, as in the experiments, have a higher correlation coefficient than the same evaluator on different test suites with the same MT evaluation measure of intelligibility. Thus, the difficulty and size of the test suite is another key factor in MT evaluation. The following is further analysis of the influence of test suites.

3.2.2 Influence of the Test Suite

For the different parts of the test suite, the researchers have the discriminability and difficulty of intelligibility calculated using equations (3) and (4), which can give one a hint of the reason for their influence on the MT evaluation results.

Table 2. Discriminability and difficulty of test suites with intelligibility by different evaluators. The 300 sentences in the test suite are divided into 3 parts and evaluated with intelligibility separately.

Sentences	Evaluator	Discriminability	Difficulty
1-100	A	0.23	0.50
1-100	B	0.31	0.44
101-200	A	0.23	0.56
101-200	B	0.31	0.62
201-300	A	0.24	0.43
201-300	B	0.34	0.53
All 300	A	0.23	0.50
All 300	B	0.32	0.53

From Table 2, one can see that different parts of a test set may have different difficulty and discriminability levels. Since all evaluation tasks need better discriminability capability, the evaluator needs to pick out proper test sentences for the evaluation task. Taking evaluator A as an example, the difficulty of different parts of the test suites are 0.50 for sentences 1-100, 0.56 for sentences 101-200, and 0.43 for sentences 201-300. The different difficulty levels led to different correlation coefficients between different parts of the test suites. For example, sentences 101-200 and 201-300 differ greatly in difficulty, and the difference in correlation coefficients is also lower in Table 1 (only 0.945). Another factor found in Table 2 is that the results of evaluators A and B have different discriminability, the former about 0.23, and the latter 0.32. That means their evaluation score has a different distribution style. In fact, this

phenomenon has a vital influence on the correlation coefficient of two evaluation results, which is highly related to the evaluation result.

The above study of the evaluation performance is made on a public-available Japanese test suite. One does have to notice that the evaluation performance measures are language-independent, which ensures the applicability of the method to the Chinese language, or other language pairs.

To study other performance measures, a test on a Chinese suite is made below.

As described above, besides the difficulty and discriminability, another key factor for the test suite is the size. The larger the size of the test suite, the more stable and reliable the MT evaluation result becomes. Taking the popular automatic evaluation methods of BLEU and GTM as example, the influence of the size of the test suite, *i.e.* the number of sentences it contains, is tested using the 863 National High-tech Program MT evaluation corpus. This corpus is widely used for the evaluation of MT systems in mainland China. The corpus contains 1019 sentences. An experiment was carried out on the BLEU and GTM methods to test the influence of the size of a test suite for an English-to-Chinese translation system. The result is shown in Figure 1.

When the test suite is small, *i.e.* there is small number of sentences in the test suite, the MT evaluation score fluctuates violently. While when the test suite contains more than 80 sentences, the fluctuation becomes less violent and goes flat after 400 sentences. Figure 1 shows that the two methods have similar tendencies, which shows that they have similar demands of the test suite size.

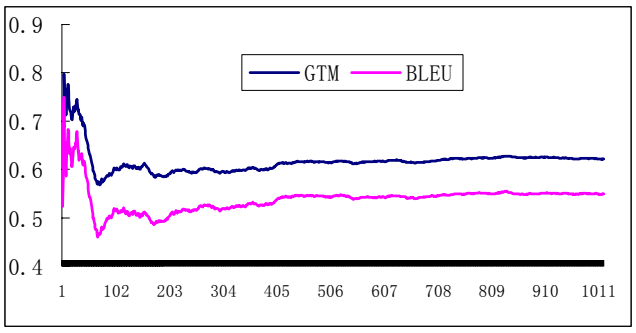
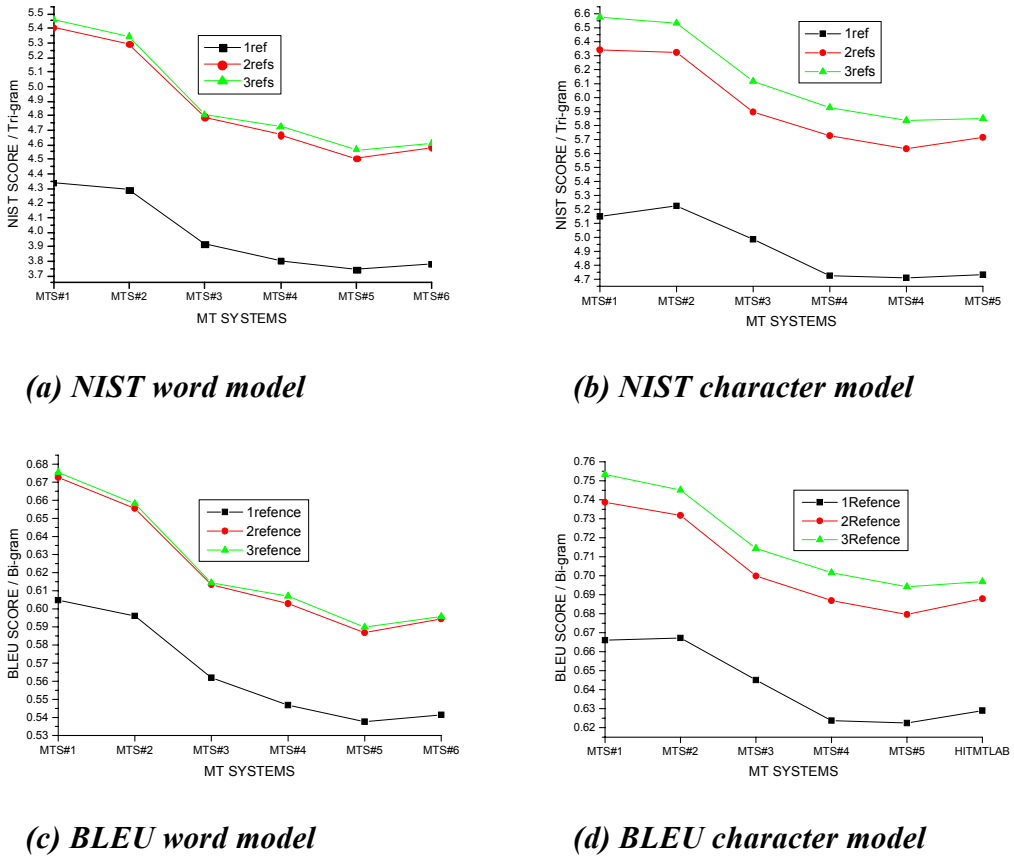


Figure 1. MT evaluation score changes with the increasing of sentence in the test corpus. The score stabilizes when the corpus contains more than 400 sentences. The experiment is made on an English-to-Chinese MT system.

Another aspect of the influence of the size of test suite can be revealed by the number of reference translations in NIST and BLEU evaluation. To get a higher quality of evaluation result, the BLEU and NIST methods can have multiple reference translations. Figure 2 shows the influence of the number of reference translations on BLEU and NIST evaluation results.



(a) NIST word model ***(b) NIST character model***

(c) BLEU word model ***(d) BLEU character model***

Figure 2. Evaluation results with different number of reference translations by (a) NIST word model, (b) NIST character model, (c) BLEU word model and (d) BLEU character model for 6 English-Chinese MT systems. The word model calculates the MT evaluation scores in terms of Chinese words, while the character model is in terms of Chinese characters.

The BLEU and NIST evaluations are implemented with two different language models: The character model, which takes Chinese characters as unit of scoring, while the word model takes the Chinese word as the unit. The Chinese sentences are segmented into words by a Chinese segmentor (which was developed at Harbin Institute of Technology, http://ir.hit.edu.cn). In BLEU and NIST evaluation, one can see that the scores go up with the increasing number of reference translations. Compared to the character model, the word model scores saturate faster with an increasing number of references, which means it has a lower demand for references. This is also the case for the BLEU models. A possible reason for this phenomenon is that a word is not easy to be matched in extra-translation reference, while new characters come out even after a big number of references. This experiment gives researchers a hint that synonyms can improve the performance of similarity-based MT evaluation methods such as BLEU and NIST.

4. Visualization of MT evaluation scores and system clustering

MT evaluation has been extensively studied in recent years. However, the various MT evaluation methods just render a score for each system or translation sentence. The score scales also vary among methods. The BLEU and GTM score has a value between 0 and 1. NIST has a lower bound of 0 with no upper bound. The manual evaluation of fidelity and accuracy usually has discrete quality levels. This makes it quite ambiguous to understand the meanings of the scores. This section intends to make it easier to understand the MT evaluation scores by visualizing the scores of evaluation results.

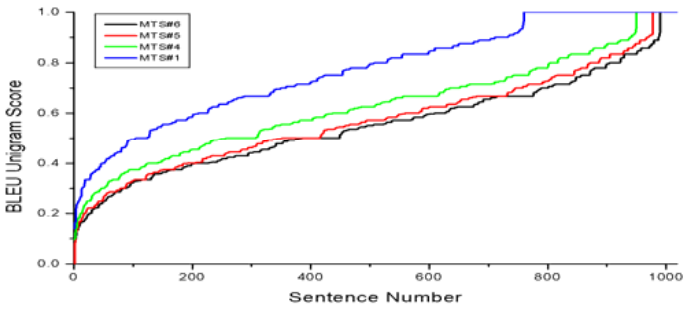
4.1 Visualization of MT Evaluation Scores

The BLEU and NIST evaluation methods have been popular in MT evaluation research. This research project makes MT evaluation experiments using these methods for a better understanding of the result. The MT evaluation data is visualized in the diagram as shown in Figure 4. Figure 4 exhibits the MT evaluation results with the test suite of 1019 sentences selected from the 863 National High-tech Program MT Evaluation corpus for machine translation, as introduced in section 3.2. Four systems are evaluated with the BLEU method. The diagram is produced with the algorithm in Figure 3.

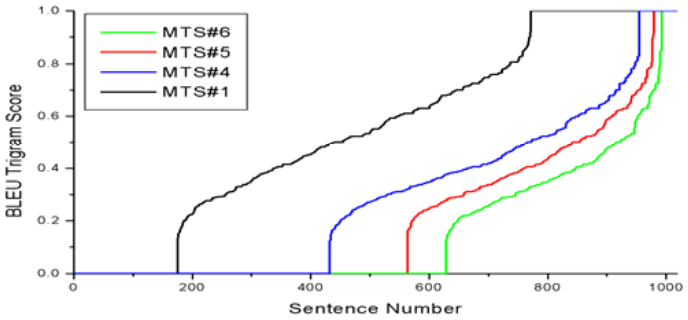
Algorithm: Visualization of system scores by plotting lines in a diagram

```
1:  INPUT: T←{Ti: t∈Ti, t is a translation by MT system MTSi}
2:  //Process the MT translation and get the BLEU scores
3:  For each machine translation system MTSi do
4:  For each translation t∈Ti by machine translation systems MTSi do
5:      Score{t}←{sti| sti is the BLEU score of the translation ti}
6:  End for
7:  //Plot a line of the BLEU scores for each MT system
8:  Score{t}←Score{t} {the BLEU scores sorted in ascending order}
9:  For i=1 to |T| {number of items in the translation set T} do
10:     Plot a point (i,sti) in the diagram
11: End for
12: End for
13: Output: a diagram in which every MT system is presented with a curve
```

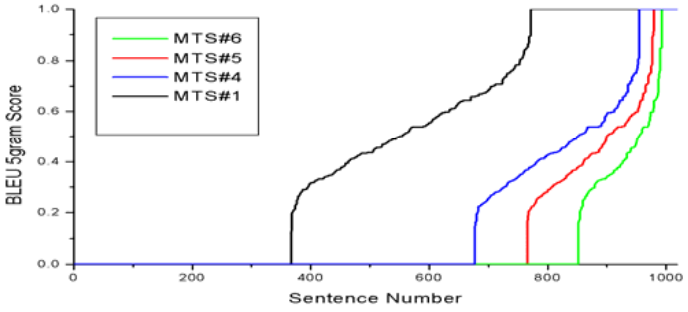
Figure 3. Algorithm: Visualization of system scores by plotting lines in a diagram



(a) BLEU 1-gram



(b) BLEU 3-gram



(c) BLEU 5-gram

Figure 4. Machine translation evaluation scores of 4 MT systems on 1019 sentences with (a) 1-gram, (b) 3-gram and (c) 5-gram BLEU method. Each line manifests the quality performance of a MT system. A line on the left and upper stands for a system with higher translation quality.

Figure 4 is the diagram from the algorithm for visualization of system scores, in which each system is represented by a line drawn according to the scores of the translations. From the lines of MT systems, one can draw the following conclusions about the MT evaluation performance. 1) The longer the N-gram, the more difficult the test is, and the lower the scores obtained by MT systems. The lines in the diagram shift to the right side when the N-gram shifts from unigram to 5-gram. The leftmost line represents the performance of the best system. 2) The gap between the lines changes with the difficulty of the test. As seen in the Figure 4(a) of the unigram scores, the lines representing systems #2, #3, and #4 are very near to each other, while the gap becomes much larger between the trigram lines in Figure 4(b). This is because the difficulty of the test influences the discriminability of the evaluation.

The visualization method is based on NIST, BLEU or a similar MT evaluation score, but is more intuitional and easier to understand. On the one hand, the evaluation is not only presented for the whole system, but also each translation; on the other hand, the tendency of the lines manifests the quality characteristics of MT systems, while the gap represents the difference. From the diagram, one can directly see the difficulty and discriminability of the MT evaluation. This has fully taken advantage of the diagrams over pure numbers.

4.2 System Clustering Based on Various MT Evaluation Scores

The above section presents a diagram presenting the evaluation scores of the MT systems, which shows the translation quality of several systems. To make the quality difference clearer, system clustering is utilized for visualizing the distances of MT systems in respect to translation quality in this section. This process involves calculating the distances of translation quality, as shown in the algorithm of Figure 5.

The MT systems are evaluated by several manual and automatic evaluation methods. The evaluation methods are: F-measure of intelligibility and accuracy, error typology scoring ET as in [Guessoum and Zanout 2001], separate linguistic points as in [Yu 1993], BLEU word model, NIST word model, language model probability, edit distance and DICE coefficient as in [Yao *et al.* 2002]. As different evaluation methods have different value scopes, the scores as in step 3 to step 9 of the algorithm have been normalized. After the normalization, the value of MT scores varies between 0~1. The normalized scores are shown in Table 3. The clustering dendrogram is shown in Figure 6.

The methods introduced in this experiment are as follows: 1) F-measure is the F1 measure, which integrates the manual metrics of intelligibility and fidelity. 2) ET is a weighted sum of scores from different Types of Errors. 3) SLP comes from the automatic scoring based on a Separate Language Points, which measures different linguistic phenomena based on a human-edited test suite. 4) BLEUW and NISTW is the BLEU/NIST score measured on Chinese word model, which takes words instead of characters as the unit of

comparison. 5) LM is the score from a language model, specifically a bi-gram model in this article. 6) EDist is a score from edit distance between the translation and the reference. 7) DICE is a score based on the DICE coefficient of the translation and the reference.

Algorithm: Similarity histogram-based incremental MT system clustering

INPUT: Score{MTSi}←{sco_mti| BLEU scores of translations by MTSi }
// **Normalize the MT BLEU scores**
For each machine translation system MTSi **do**
max{sco_mti}←sco_mti{the maximum BLEU score in Score{MTSi}
min{sco_mti}←sco_mti{the minimum BLEU score in Score{MTSi}
For each sco_mti **do**

$$sco\_mti = \frac{sco\_mti - \min\{sco\_mti\}}{\max\{sco\_mti\} - \min\{sco\_mti\}}$$

End for
End for
//Similarity histogram-based incremental MT system clustering
L←Empty list{Cluster list}
For each MT system mts **do**
    For each cluster c in L **do**
        HRold = HRc
        Simulate adding mts to c
        If (HRnew ≧ HRold) OR ((HRnew > HRmin) AND (HRold – Hrnew < ε)) **then**
            Add mts to c
        End if
    End for
    If mts was not added to any cluster **then**
        Create a new cluster c
        Add mts to c
        Add c to L
    End if
End for
Output: a histogram of MT systems

Figure 5. Similarity histogram-based incremental MT system clustering

Table 3. Normalized scores of MT systems by various MT evaluation methods. The scores are obtained with various MT evaluation methods that have different score scopes. The scores are normalized for system clustering.

MTS	F-measure	ET	SLP	BLEUW	NISTW	LM	EDist	DICE
MTS#1	1.00	0.92	1.00	1.00	1.00	1.00	0.92	1.00
MTS#2	0.84	1.00	0.85	0.78	0.78	0.46	1.00	1.00
MTS#3	0.60	0.71	0.45	0.22	0.24	0.18	0.23	0.27
MTS#4	0.44	0.71	0.20	0.22	0.15	0.14	0.69	0.80
MTS#5	0.16	0.38	0.10	0.00	0.00	0.00	0.00	0.00
MTS#6	0.00	0.00	0.00	0.11	0.03	0.11	0.08	0.20

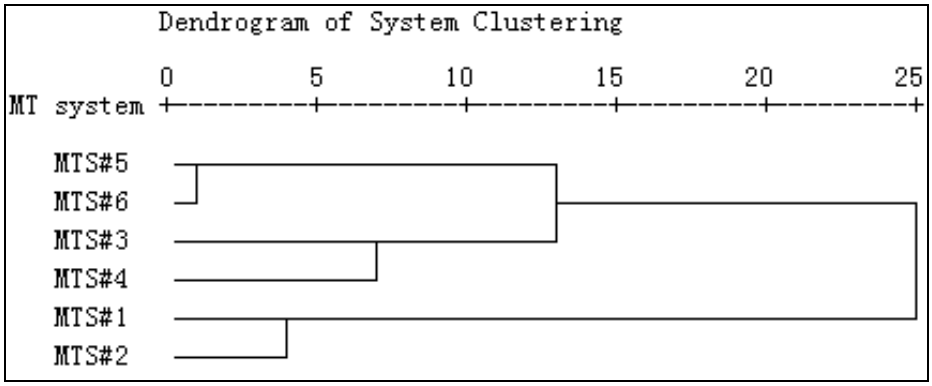


Figure 6. Cluster chart and distance between clusters of 6 MT systems. Systems are clustered according to their quality difference.

The cluster chart in the dendrogram in Figure 6 is a clear representation of the machine translation system quality. As seen from this dendrogram, the systems MTS#5 and MTS#6 are very similar to each other and are clustered first. The MTS#1 and MTS#2 have a second smallest difference. After MTS#3 and MTS#4 are clustered as one, the clustering goes on, and all the systems cluster into a binary tree. This clustering dendrogram is an easy way for a clear presentation of MT system quality based on ensemble of various evaluation scores.

5. Conclusion

This paper is an effort towards MT evaluation performance analysis and better rendering of MT evaluation results. After a general framework is proposed for the description of MT evaluation measure and the test suite, some instances are given including whether the automatic measure is consistent with human evaluation, whether MT evaluation results from various measures or test suites are consistent, whether the content of the test suite is suitable

for performance evaluation, the degree of difficulty of the test suite and its influence on the MT evaluation, the relationship of MT evaluation result significance and the size of the test suite, etc. For better clarification of the framework, a visualization method is introduced for presenting the results. The MT evaluation performance analysis can help a lot in designing test suites for different MT evaluation methods. The visualization method, on the one hand, gives an intuitive representation of the quality difference of MT systems; on the other hand, it is an easy way to assemble of the different evaluation results.

Acknowledgements

The research project is supported by the High-Tech Research and Development Program of Jiangsu Province China (Contract No. GB2005020, BK2006539), the Natural Science Foundation for Higher Education in Jiangsu Province (Contract No. 06KJB520095), Natural Science Foundation of Guangdong Province (Contract No. 108B6040600).

References

ALPAC, "Languages and machines: computers in translation and linguistics," A report by the Automatic Language Processing Advisory Committee, National Research Council. Washington, D.C., National Academy of Sciences, 1966.

Brew, C., and H.S. Thompson, "Automatic evaluation of computer generated text: a progress report on the TextEval project," In *Proceedings of the Human Language Technology Workshop*, 1994, pp. 108-113.

Darwin, M., "Trial and Error: An Evaluation Project on Japanese English MT Output Quality," In *Proceedings of the MT Summit*,2001, Santiago de Compostela, Galicia, Spain, pp.57-63.

Doyon, J., K. Taylor, and J. White, "The DARPA Machine Translation Evaluation Methodology: Past and Present," In *Proceedings of the AMTA*, 1998, Philadelphia, PA.

Forner, M., and J. White, "Predicting MT fidelity from noun-compound handling," In *Proceedings of the Workshop MT Evaluation: Who Did What To Whom held in conjunction with Machine Translation Summit VIII*, 2001, Santiago de Compostela, Spain, pp.45-48.

Guessoum, A., and R. Zantout, "Semi-automatic evaluation of the grammatical coverage of machine translation systems," In *Proceedings of the MT Summit Conference*, 2001, Santiago de Compostela, pp.133-138.

ISLE, "The ISLE classification of machine translation evaluations, draft 1," A document by the International Standards for Language Engineering, http://www.isi.edu/natural-language/mteval/, 2000.

Jones, S., and J. Galliers, "Evaluating Natural Language Processing Systems," Technical Report 291, University of Cambridge Computer Laboratory,1993.

Keiji, Y., F. Sugaya, T. Takezawa, S. Yamamoto,and M. Yanagida, "An automatic evaluation method of translation quality using translation answer candidates queried from a parallel corpus," In *Proceedings of MT Summit Conference*, 2001, Santiago de Compostela, pp.373-378.

Koh, S., J. Maeng, J. Y. LEE, Y. S. CHAE, and K. S. Choi, "A test suite for evaluation of English-to-Korean machine translation systems," In *Proceedings of MT Summit Conference*, 2001, Santiago de Compostela.

Melamed, I.D., R.Green, and J.P.Turian, "Precision and recall of machine translation," In *Proceedings of the NAACL/Human Language Technology*, 2003, Edmonton, Canada.

Niβen, S., F. J. Och, G. Leusch, and H. Ney, "An evaluation tool for machine translation: fast evaluation for MT research," In *Proceedings of the 2nd International Conference on Language Resources and Evaluation*, 2000, Athens, Greece, pp.39-45.

NIST, "The NIST 2002 machine translation evaluation plan," A document by the National Institute of Standards and Technology, http://www.nist.gov/speech/ tests/mt/doc/2002-MT-EvalPlan- v1.3.pdf. 2002.

Papineni, K., S.Roukos, T.Ward, and W.-J. Zhu, "BLEU: a method for automatic evaluation of MT," Research Report, Computer Science RC22176(W0109-022), IBM Research Division, T.J.Watson Research Center, 2001.

Popescu-Belis, A., "Evaluation of natural language processing systems: a model for coherence verification of quality measure," In Marc Blasband and Patrick Paroubek, editors, A Blueprint for a General Infrastructure for Natural Language Processing Systems Evaluation Using Semi-Automatic Quantitative Black Box Approach in a Multilingual Environment. ELSE Project LE4-8340 (Evaluation in Language and Speech Engineering), 1999.

Wang, X., "Education Measurement," East China Normal University Press, 2001, pp. 129~161. (in Chinese)

Yao, J., M. Zhou, H. Yu, T. Zhao, and S. Li, "An Automatic Evaluation Method for Localization Oriented Lexicalised EBMT System," In *Proceedings of the 19th Conference on Computational Linguistics (COLING'2002)*, August 24, 2002, TaiPei, Taiwan, pp.1142-1148.

Yasuhiro, A., K. Imamura, and E. Sumita, "Using multiple edit distances to automatically rank machine translation output," In *Proceedings of the MT Summit Conference*, 2001, Santiago de Compostela, pp. 15-20.

Yokoyama, S., H. Kashioka, A. Kumano, M. Matsudaira, Y. Shirokizawa, M. Kawagoe, S. Kodama, H. Kashioka, T. Ehara, S. Miyazawa, and Y. Nakajima, "Quantitative evaluation of machine translation using two-way MT," In *Proceeding of Machine Translation Summit VII*, 1999, pp.568--573.

Yu, S., "Automatic Evaluation of Quality for Machine Translation Systems," *Machine Translation*, 8, 1993, pp.117-126.

Appendix

This section presents the human evaluation results from [Darwin 2001] on eight English-to-Japanese MT systems. Two popular metrics are used in the human evaluation: intelligibility and accuracy. The evaluators score the systems on a 5 point scale.

Table A1. Overall English-to-Japanese Average Scores (Possible Score from 1 to 5 Points).

Metrics	EJsys-1	EJsys-2	EJsys-3	EJsys-4	EJsys-5	EJsys-6	EJsys-7	EJsys-8
Intelligibility	2.33	3.39	3.42	3.32	3.00	3.01	3.11	2.87
Accuracy	2.42	3.60	3.62	3.45	3.13	3.15	3.27	2.99

Table A2. E-to-J Average Scores by Evaluator A and B (phase by phase), the column "I" lists intelligibility scores, and A column lists accuracy scores.

Test Suite	EJsys-1		EJsys-2		EJsys-3		EJsys-4		EJsys-5		EJsys-6		EJsys-7		EJsys-8	
	I	A	I	A	I	A	I	A	I	A	I	A	I	A	I	A
Sent#1-100(A)	2.38	2.62	3.25	3.56	3.30	3.54	3.14	3.48	3.10	3.29	2.97	3.26	3.08	3.33	2.81	3.04
Sent#101-200(A)	2.67	2.83	3.53	3.87	3.58	3.91	3.32	3.65	3.17	3.45	3.17	3.53	3.33	3.69	3.14	3.43
Sent#201-300(A)	2.11	2.41	3.02	3.54	3.05	3.61	3.01	3.40	2.67	3.06	2.71	3.02	2.65	3.07	2.56	2.86
All 300(A)	2.39	2.62	3.27	3.66	3.31	3.69	3.16	3.51	2.98	3.27	2.95	3.27	3.02	3.36	2.84	3.11
Sent#1-100(B)	1.91	1.76	3.15	3.08	3.08	2.98	3.08	2.87	2.73	2.55	2.78	2.65	2.83	2.75	2.48	2.39
Sent#101-200(B)	2.65	2.60	3.86	3.86	3.89	3.90	3.74	3.60	3.32	3.29	3.42	3.35	3.59	3.53	3.31	3.22
Sent#201-300(B)	2.25	2.29	3.50	3.66	3.61	3.77	3.60	3.68	3.03	3.15	3.02	3.09	3.20	3.25	2.89	2.97
All 300(B)	2.27	2.22	3.50	3.53	3.53	3.55	3.47	3.38	3.03	3.00	3.07	3.03	3.21	3.18	2.89	2.86