Proceedings of the 41st Annual Meeting of the Association
for Computational Linguistics, July 2003, pp. 272-279.

Improved Source-Channel Models for Chinese Word Segmentation[1]

Jianfeng Gao, Mu Li and Chang-Ning Huang
Microsoft Research, Asia
Beijing 100080, China
{jfgao, t-muli, cnhuang}@microsoft.com

Abstract

This paper presents a Chinese word segmentation system that uses improved source-channel models of Chinese sentence generation. Chinese words are defined as one of the following four types: lexicon words, morphologically derived words, factoids, and named entities. Our system provides a unified approach to the four fundamental features of word-level Chinese language processing: (1) word segmentation, (2) morphological analysis, (3) factoid detection, and (4) named entity recognition. The performance of the system is evaluated on a manually annotated test set, and is also compared with several state-of-the-art systems, taking into account the fact that the definition of Chinese words often varies from system to system.

1 Introduction

Chinese word segmentation is the initial step of many Chinese language processing tasks, and has attracted a lot of attention in the research community. It is a challenging problem due to the fact that there is no standard definition of Chinese words.

In this paper, we define Chinese words as one of the following four types: entries in a lexicon, morphologically derived words, factoids, and named entities. We then present a Chinese word segmentation system which provides a solution to the four fundamental problems of word-level Chinese language processing: word segmentation, morphological analysis, factoid detection, and named entity recognition (NER).

There are no word boundaries in written Chinese text. Therefore, unlike English, it may not be desirable to separate the solution to word segmentation from the solutions to the other three problems. Ideally, we would like to propose a unified approach to all the four problems. The unified approach we used in our system is based on the improved source-channel models of Chinese sentence generation, with two components: a source model and a set of channel models. The source model is used to estimate the generative probability of a word sequence, in which each word belongs to one word type. For each word type, a channel model is used to estimate the generative probability of a character string given the word type. So there are multiple channel models. We shall show in this paper that our models provide a statistical framework to corporate a wide variety linguistic knowledge and statistical models in a unified way.

We evaluate the performance of our system using an annotated test set. We also compare our system with several state-of-the-art systems, taking into account the fact that the definition of Chinese words often varies from system to system.

In the rest of this paper: Section 2 discusses previous work. Section 3 gives the detailed definition of Chinese words. Sections 4 to 6 describe in detail the improved source-channel models. Section 8 describes the evaluation results. Section 9 presents our conclusion.

2 Previous Work

Many methods of Chinese word segmentation have been proposed: reviews include (Wu and Tseng, 1993; Sproat and Shih, 2001). These methods can be roughly classified into dictionary-based methods and statistical-based methods, while many state-of-the-art systems use hybrid approaches.

In dictionary-based methods (e.g. Cheng et al., 1999), given an input character string, only words that are stored in the dictionary can be identified. The performance of these methods thus depends to

[1] We would like to thank Ashley Chang, Jian-Yun Nie, Andi Wu and Ming Zhou for many useful discussions, and for comments on earlier versions of this paper. We would also like to thank Xiaoshan Fang, Jianfeng Li, Wenfeng Yang and Xiaodan Zhu for their help with evaluating our system.

a large degree upon the coverage of the dictionary, which unfortunately may never be complete because new words appear constantly. Therefore, in addition to the dictionary, many systems also contain special components for unknown word identification. In particular, statistical methods have been widely applied because they utilize a probabilistic or cost-based scoring mechanism, instead of the dictionary, to segment the text. These methods however, suffer from three drawbacks. First, some of these methods (e.g. Lin et al., 1993) identify unknown words without identifying their types. For instance, one would identify a string as a unit, but not identify whether it is a person name. This is not always sufficient. Second, the probabilistic models used in these methods (e.g. Teahan et al., 2000) are trained on a segmented corpus which is not always available. Third, the identified unknown words are likely to be linguistically implausible (e.g. Dai et al., 1999), and additional manual checking is needed for some subsequent tasks such as parsing.

We believe that the identification of unknown words should not be defined as a separate problem from word segmentation. These two problems are better solved simultaneously in a unified approach. One example of such approaches is Sproat et al. (1996), which is based on weighted finite-state transducers (FSTs). Our approach is motivated by the same inspiration, but is based on a different mechanism: the improved source-channel models. As we shall see, these models provide a more flexible framework to incorporate various kinds of lexical and statistical information. Some types of unknown words that are not discussed in Sproat's system are dealt with in our system.

3 Chinese Words

There is no standard definition of Chinese words – linguists may define words from many aspects (e.g. Packard, 2000), but none of these definitions will completely line up with any other. Fortunately, this may not matter in practice because the definition that is most useful will depend to a large degree upon how one uses and processes these words.

We define Chinese words in this paper as one of the following four types: (1) entries in a lexicon (*lexicon words* below), (2) morphologically derived words, (3) factoids, and (4) named entities, because these four types of words have different functionalities in Chinese language processing, and are processed in different ways in our system. For example, the plausible word segmentation for the sentence in Figure 1(a) is as shown. Figure 1(b) is the output of our system, where words of different types are processed in different ways:

(a) 朋友们/十二点三十分/高高兴兴/到/李俊生/教授/家/吃饭 (Friends happily go to professor Li Junsheng's home for lunch at twelve thirty.)

(b) [朋友+们 MA_S] [十二点三十分 12:30 TIME] [高兴 MR_AABB] [到] [李俊生 PN] [教授] [家] [吃饭]

Figure 1: (a) A Chinese sentence. Slashes indicate word boundaries. (b) An output of our word segmentation system. Square brackets indicate word boundaries. + indicates a morpheme boundary.

- For lexicon words, word boundaries are detected.
- For morphologically derived words, their morphological patterns are detected, e.g. 朋友们 'friend+s' is derived by affixation of the plural affix 们 to the noun 朋友 (MA_S indicates a suffixation pattern), and 高高兴兴 'happily' is a reduplication of 高兴 'happy' (MR_AABB indicates an AABB reduplication pattern).
- For factoids, their types and normalized forms are detected, e.g. 12:30 is the normalized form of the time expression 十二点三十分 (TIME indicates a time expression).
- For named entities, their types are detected, e.g. 李俊生 'Li Junsheng' is a person name (PN indicates a person name).

In our system, we use a unified approach to detecting and processing the above four types of words. This approach is based on the improved source-channel models described below.

4 Improved Source-Channel Models

Let S be a Chinese sentence, which is a character string. For all possible word segmentations W, we will choose the most likely one W^* which achieves the highest conditional probability $P(W|S)$: $W^* = \text{argmax}_W P(W|S)$. According to Bayes' decision rule and dropping the constant denominator, we can equivalently perform the following maximization:

$$W^* = \arg\max_W P(W)P(S \mid W) . \quad (1)$$

Following the Chinese word definition in Section 3, we define word class C as follows: (1) Each lexicon

Word class	Class model	Linguistic Constraints
Lexicon word (LW)	*P(S|LW)*=1 if *S* forms a word lexicon entry, 0 otherwise.	Word lexicon
Morphologically derived word (MW)	*P(S|MW)*=1 if *S* forms a morph lexicon entry, 0 otherwise.	Morph-lexicon
Person name (PN)	Character bigram	family name list, Chinese PN patterns
Location name (LN)	Character bigram	LN keyword list, LN lexicon, LN abbr. list
Organization name (ON)	Word class bigram	ON keyword list, ON abbr. list
Transliteration names (FN)	Character bigram	transliterated name character list
Factoid[2] (FT)	*P(S|FT)*=1 if *S* can be parsed using a factoid grammar *G*, 0 otherwise	Factoid rules (presented by FSTs).

Figure 2. Class models

word is defined as a class; (2) each morphologically derived word is defined as a class; (3) each type of factoids is defined as a class, e.g. all time expressions belong to a class TIME; and (4) each type of named entities is defined as a class, e.g. all person names belong to a class PN. We therefore convert the word segmentation *W* into a word class sequence *C*. Eq. 1 can then be rewritten as:

$$C^* = \arg\max_C P(C)P(S \mid C) . \quad (2)$$

Eq. 2 is the basic form of the source-channel models for Chinese word segmentation. The models assume that a Chinese sentence *S* is generated as follows: First, a person chooses a sequence of concepts (i.e., word classes *C*) to output, according to the probability distribution *P*(*C*); then the person attempts to express each concept by choosing a sequence of characters, according to the probability distribution *P*(*S*|*C*).

The source-channel models can be interpreted in another way as follows: *P*(*C*) is a stochastic model estimating the probability of word class sequence. It indicates, given a context, how likely a word class occurs. For example, person names are more likely to occur before a title such as 教授 'professor'. So *P*(*C*) is also referred to as *context model* afterwards. *P*(*S*|*C*) is a generative model estimating how likely a character string is generated given a word class. For example, the character string 李俊生 is more likely to be a person name than 里俊生 'Li Junsheng' because 李 is a common family name in China while 里 is not. So *P*(*S*|*C*) is also referred to as *class model* afterwards. In our system, we use the improved source-channel models, which contains one context model (i.e., a trigram language model in our case) and a set of class models of different types, each of which is for one class of words, as shown in Figure 2.

Although Eq. 2 suggests that class model probability and context model probability can be combined through simple multiplication, in practice some weighting is desirable. There are two reasons. First, some class models are poorly estimated, owing to the sub-optimal assumptions we make for simplicity and the insufficiency of the training corpus. Combining the context model probability with poorly estimated class model probabilities according to Eq. 2 would give the context model too little weight. Second, as seen in Figure 2, the class models of different word classes are constructed in different ways (e.g. name entity models are *n*-gram models trained on corpora, and factoid models are compiled using linguistic knowledge). Therefore, the quantities of class model probabilities are likely to have vastly different dynamic ranges among different word classes. One way to balance these probability quantities is to add several *class model weight CW*, each for one word class, to adjust the class model probability *P*(*S*|*C*) to $P(S|C)^{CW}$. In our experiments, these class model weights are determined empirically to optimize the word segmentation performance on a development set.

Given the source-channel models, the procedure of word segmentation in our system involves two steps: First, given an input string *S*, all word candidates are generated (and stored in a lattice). Each candidate is tagged with its word class and the class

[2] In our system, we define ten types of factoid: date, time (TIME), percentage, money, number (NUM), measure, e-mail, phone number, and WWW.

model probability $P(S'|C)$, where S' is any substring of S. Second, Viterbi search is used to select (from the lattice) the most probable word segmentation (i.e. word class sequence C^*) according to Eq. (2).

5 Class Model Probabilities

Given an input string S, all class models in Figure 2 are applied simultaneously to generate word class candidates whose class model probabilities are assigned using the corresponding class models:

- Lexicon words: For any substring $S' \subseteq S$, we assume $P(S'|C) = 1$ and tagged the class as lexicon word if S' forms an entry in the word lexicon, $P(S'|C) = 0$ otherwise.
- Morphologically derived words: Similar to lexicon words, but a morph-lexicon is used instead of the word lexicon (see Section 5.1).
- Factoids: For each type of factoid, we compile a set of finite-state grammars G, represented as FSTs. For all $S' \subseteq S$, if it can be parsed using G, we assume $P(S'|FT) = 1$, and tagged S' as a factoid candidate. As the example in Figure 1 shows, 十二点三十分 is a factoid (time) candidate with the class model probability P(十二点三十分|TIME) =1, and 十二 and 三十 are also factoid (number) candidates, with P(十二|NUM) = P(三十|NUM) =1
- Named entities: For each type of named entities, we use a set of grammars and statistical models to generate candidates as described in Section 5.2.

5.1 Morphologically derived words

In our system, the morphologically derived words are generated using five morphological patterns: (1) **affixation**: 朋友们 (friend - plural) 'friends'; (2) **reduplication**: 高兴 'happy' → 高高兴兴 'happily'; (3) **merging**: 上班 'on duty' + 下班 'off duty' →上下班 'on-off duty'; (4) **head particle** (i.e. expressions that are verb + comp): 走 'walk' + 出去 'out' → 走出去 'walk out'; and (5) **split** (i.e. a set of expressions that are separate words at the syntactic level but single words at the semantic level): 吃了饭 'already ate', where the bi-character word 吃饭 'eat' is split by the particle 了 'already'.

It is difficult to simply extend the well-known techniques for English (i.e., finite-state morphology) to Chinese due to two reasons. First, Chinese morphological rules are not as 'general' as their English counterparts. For example, English plural nouns can be in general generated using the rule 'noun + s → plural noun'. But only a small subset of Chinese nouns can be pluralized (e.g. 朋友们) using its Chinese counterpart 'noun + 们 → plural noun' whereas others (e.g. 南瓜 'pumpkin**s**') cannot. Second, the operations required by Chinese morphological analysis such as copying in reduplication, merging and splitting, cannot be implemented using the current finite-state networks[3].

Our solution is the extended lexicalization. We simply collect all morphologically derived word forms of the above five types and incorporate them into the lexicon, called *morph lexicon*. The procedure involves three steps: **(1) Candidate generation**. It is done by applying a set of morphological rules to both the word lexicon and a large corpus. For example, the rule 'noun + 们 → plural noun' would generate candidates like 朋友们. **(2) Statistical filtering**. For each candidate, we obtain a set of statistical features such as frequency, mutual information, left/right context dependency from a large corpus. We then use an information gain-like metric described in (Chien, 1997; Gao et al., 2002) to estimate how likely a candidate is to form a morphologically derived word, and remove 'bad' candidates. The basic idea behind the metric is that a Chinese word should appear as a stable sequence in the corpus. That is, the components within the word are strongly correlated, while the components at both ends should have low correlations with words outside the sequence. **(3) Linguistic selection.** We finally manually check the remaining candidates, and construct the morph-lexicon, where each entry is tagged by its morphological pattern.

5.2 Named entities

We consider four types of named entities: person names (PN), location names (LN), organization names (ON), and transliterations of foreign names (FN). Because any character strings can be in principle named entities of one or more types, to limit the number of candidates for a more effective search, we generate named entity candidates, given an input string, in two steps: First, for each type, we use a set of constraints (which are compiled by

[3] Sproat et al. (1996) also studied such problems (with the same example) and uses weighted FSTs to deal with the affixation.

linguists and are represented as FSTs) to generate only those 'most likely' candidates. Second, each of the generated candidates is assigned a class model probability. These class models are defined as generative models which are respectively estimated on their corresponding named entity lists using maximum likelihood estimation (MLE), together with smoothing methods[4]. We will describe briefly the constraints and the class models below.

5.2.1 Chinese person names

There are two main constraints. (1) PN patterns: We assume that a Chinese PN consists of a family name **F** and a given name **G**, and is of the pattern **F+G**. Both **F** and **G** are of one or two characters long. (2) Family name list: We only consider PN candidates that begin with an **F** stored in the family name list (which contains 373 entries in our system).

Given a PN candidate, which is a character string *S'*, the class model probability $P(S'|PN)$ is computed by a character bigram model as follows: (1) Generate the family name sub-string S_F, with the probability $P(S_F|F)$; (2) Generate the given name sub-string S_G, with the probability $P(S_G|G)$ (or $P(S_{G1}|G_1)$); and (3) Generate the second given name, with the probability $P(S_{G2}|S_{G1},G_2)$. For example, the generative probability of the string 李俊生 given that it is a PN would be estimated as P(李俊生|PN) = P(李|*F*)P(俊|G_1)P(生|俊,G_2).

5.2.2 Location names

Unlike PNs, there are no patterns for LNs. We assume that a LN candidate is generated given *S'* (which is less than 10 characters long), if one of the following conditions is satisfied: (1) *S'* is an entry in the LN list (which contains 30,000 LNs); (2) *S'* ends in a keyword in a 120-entry LN keyword list such as 市 'city'[5]. The probability $P(S'|LN)$ is computed by a character bigram model.

Consider a string 乌苏里江 'Wusuli river'. It is a LN candidate because it ends in a LN keyword 江 'river'. The generative probability of the string given it is a LN would be estimated as P(乌苏里江|LN) = P(乌|<*LN*>) P(苏|乌) P(里|苏) P(江|里) P(</*LN*>|江), where <*LN*> and </*LN*> are symbols denoting the beginning and the end of a LN, respectively.

5.2.3 Organization names

ONs are more difficult to identify than PNs and LNs because ONs are usually nested named entities. Consider an ON 中国国际航空公司 'Air China Corporation'; it contains an LN 中国 'China'.

Like the identification of LNs, an ON candidate is only generated given a character string *S'* (less than 15 characters long), if it ends in a keyword in a 1,355-entry ON keyword list such as 公司 'corporation'. To estimate the generative probability of a nested ON, we introduce word class segmentations of *S'*, *C*, as hidden variables. In principle, the ON class model recovers $P(S'|ON)$ over all possible *C*: $P(S'|ON) = \sum_C P(S',C|ON) = \sum_C P(C|ON)P(S'|C,ON)$. Since $P(S'|C,ON) = P(S'|C)$, we have $P(S'|ON) = \sum_C P(C|ON)\ P(S'|C)$. We then assume that the sum is approximated by a single pair of terms $P(C^*|ON)P(S'|C^*)$, where C^* is the most probable word class segmentation discovered by Eq. 2. That is, we also use our system to find C^*, but the source-channel models are estimated on the ON list.

Consider the earlier example. Assuming that C^* = LN/国际/航空/公司, where 中国 is tagged as a LN, the probability $P(S'|ON)$ would be estimated using a word class bigram model as: P(中国国际航空公司|*ON*) ≈ P(LN/国际/航空/公司|*ON*) P(中国|LN) = P(LN|<*ON*>)P(国际|LN)P(航空|国际)P(公司|航空) P(</*ON*>|公司)P(中国|LN), where P(中国|LN) is the class model probability of 中国 given that it is a LN, <*ON*> and </*ON*> are symbols denoting the beginning and the end of a ON, respectively.

5.2.4 Transliterations of foreign names

As described in Sproat et al. (1996): FNs are usually transliterated using Chinese character strings whose sequential pronunciation mimics the source language pronunciation of the name. Since FNs can be of any length and their original pronunciation is effectively unlimited, the recognition of such names is tricky. Fortunately, there are only a few hundred Chinese characters that are particularly common in transliterations.

Therefore, an FN candidate would be generated given *S'*, if it contains only characters stored in a transliterated name character list (which contains

[4] The detailed description of these models are in Sun et al. (2002), which also describes the use of cache model and the way the abbreviations of LN and ON are handled.

[5] For a better understanding, the constraint is a simplified version of that used in our system.

618 Chinese characters). The probability $P(S'|FN)$ is estimated using a character bigram model. Notice that in our system a FN can be a PN, a LN, or an ON, depending on the context. Then, given a FN candidate, three named entity candidates, each for one category, are generated in the lattice, with the class probabilities $P(S'|PN)=P(S'|LN)=P(S'|ON)=P(S'|FN)$. In other words, we delay the determination of its type until decoding where the context model is used.

6 Context Model Estimation

This section describes the way the class model probability $P(C)$ (i.e. trigram probability) in Eq. 2 is estimated. Ideally, given an annotated corpus, where each sentence is segmented into words which are tagged by their classes, the trigram word class probabilities can be calculated using MLE, together with a backoff schema (Katz, 1987) to deal with the sparse data problem. Unfortunately, building such annotated training corpora is very expensive.

Our basic solution is the bootstrapping approach described in Gao et al. (2002). It consists of three steps: (1) Initially, we use a greedy word segmentor[6] to annotate the corpus, and obtain an initial context model based on the initial annotated corpus; (2) we re-annotate the corpus using the obtained models; and (3) re-train the context model using the re-annotated corpus. Steps 2 and 3 are iterated until the performance of the system converges.

In the above approach, the quality of the context model depends to a large degree upon the quality of the initial annotated corpus, which is however not satisfied due to two problems. First, the greedy segmentor cannot deal with the segmentation ambiguities, and even after iterations, these ambiguities can only be partially resolved. Second, many factoids and named entities cannot be identified using the greedy word segmentor which is based on the dictionary.

To solve the first problem, we use two methods to resolve segmentation ambiguities in the initial segmented training data. We classify word segmentation ambiguities into two classes: overlap ambiguity (OA), and combination ambiguity (CA). Consider a character string ABC, if it can be segmented into two words either as AB/C or A/BC depending on different context, ABC is called an overlap ambiguity string (OAS). If a character string AB can be segmented either into two words, A/B, or as one word depending on different context. AB is called a combination ambiguity string (CAS). To resolve OA, we identify all OASs in the training data and replace them with a single token <OAS>. By doing so, we actually remove the portion of training data that are likely to contain OA errors. To resolve CA, we select 70 high-frequent two-character CAS (e.g. 才能 'talent' and 才/能 'just able'). For each CAS, we train a binary classifier (which is based on vector space models) using sentences that contains the CAS segmented manually. Then for each occurrence of a CAS in the initial segmented training data, the corresponding classifier is used to determine whether or not the CAS should be segmented.

For the second problem, though we can simply use the finite-state machines described in Section 5 (extended by using the longest-matching constraint for disambiguation) to detect factoids in the initial segmented corpus, our method of NER in the initial step (i.e. step 1) is a little more complicated. First, we manually annotate named entities on a small subset (call *seed set*) of the training data. Then, we obtain a context model on the seed set (called *seed model*). We thus improve the context model which is trained on the initial annotated training corpus by interpolating it with the seed model. Finally, we use the improved context model in steps 2 and 3 of the bootstrapping. Our experiments show that a relatively small seed set (e.g., 10 million characters, which takes approximately three weeks for 4 persons to annotate the NE tags) is enough to get a good improved context model for initialization.

7 Evaluation

To conduct a reliable evaluation, a manually annotated test set was developed. The text corpus contains approximately half million Chinese characters that have been proofread and balanced in terms of domain, styles, and times. Before we annotate the corpus, several questions have to be answered: (1) Does the segmentation depend on a particular lexicon? (2) Should we assume a single correct segmentation for a sentence? (3) What are the evaluation criteria? (4) How to perform a fair comparison across different systems?

[6] The greedy word segmentor is based on a forward maximum matching (FMM) algorithm: It processes through the sentence from left to right, taking the longest match with the lexicon entry at each point.

	System	Word segmentation		Factoid		PN		LN		ON	
		P%	R%	P%	R%	P%	R%	P%	R%	P%	R%
1	FMM	83.7	92.7								
2	Baseline	84.4	93.8								
3	2 + Factoid	89.9	95.5	84.4	80.0						
4	3 + PN	94.1	96.7	84.5	80.0	81.0	90.0				
5	4 + LN	94.7	97.0	84.5	80.0	86.4	90.0	79.4	86.0		
6	5 + ON	96.3	97.4	85.2	80.0	87.5	90.0	89.2	85.4	81.4	65.6

Table 1: system results

As described earlier, it is more useful to define words depending on how the words are used in real applications. In our system, a lexicon (containing 98,668 lexicon words and 59,285 morphologically derived words) has been constructed for several applications, such as Asian language input and web search. Therefore, we annotate the text corpus based on the lexicon. That is, we segment each sentence as much as possible into words that are stored in our lexicon, and tag only the new words, which otherwise would be segmented into strings of one -character words. When there are multiple segmentations for a sentence, we keep only one that contains the least number of words. The annotated test set contains in total 247,039 tokens (including 205,162 lexicon/morph-lexicon words, 4,347 PNs, 5,311 LNs, 3,850 ONs, and 6,630 factoids, etc.)

Our system is measured through multiple precision-recall (P/R) pairs, and F-measures ($F_{\beta=1}$, which is defined as 2PR/(P+R)) for each word class. Since the annotated test set is based on a particular lexicon, some of the evaluation measures are meaningless when we compare our system to other systems that use different lexicons. So in comparison with different systems, we consider only the precision-recall of NER and the number of OAS errors (i.e. crossing brackets) because these measures are lexicon independent and there is always a single unambiguous answer.

The training corpus for context model contains approximately 80 million Chinese characters from various domains of text such as newspapers, novels, magazines etc. The training corpora for class models are described in Section 5.

7.1 System results

Our system is designed in the way that components such as factoid detector and NER can be 'switched on or off', so that we can investigate the relative contribution of each component to the overall word segmentation performance.

The main results are shown in Table 1. For comparison, we also include in the table (Row 1) the results of using the greedy segmentor (FMM) described in Section 6. Row 2 shows the baseline results of our system, where only the lexicon is used. It is interesting to find, in Rows 1 and 2, that the dictionary-based methods already achieve quite good recall, but the precisions are not very good because they cannot identify correctly unknown words that are not in the lexicon such factoids and named entities. We also find that even using the same lexicon, our approach that is based on the improved source-channel models outperforms the greedy approach (with a slight but statistically significant different i.e., $P < 0.01$ according to the t test) because the use of context model resolves more ambiguities in segmentation. The most promising property of our approach is that the source-channel models provide a flexible framework where a wide variety of linguistic knowledge and statistical models can be combined in a unified way. As shown in Rows 3 to 6, when components are switched on in turn by activating corresponding class models, the overall word segmentation performance increases consistently.

We also conduct an error analysis, showing that 86.2% of errors come from NER and factoid detection, although the tokens of these word types consist of only 8.7% of all that are in the test set.

7.2 Comparison with other systems

We compare our system – henceforth **SCM**, with other two Chinese word segmentation systems[7]:

[7] Although the two systems are widely accessible in mainland China, to our knowledge no standard evaluations on Chinese word segmentation of the two systems have been published by press time. More comprehensive comparisons (with other well-known systems) and detailed error analysis form one area of our future work.

System	# OAS Errors	LN			PN			ON		
		P %	R %	$F_{\beta=1}$	P %	R %	$F_{\beta=1}$	P %	R %	$F_{\beta=1}$
MSWS	63	93.5	44.2	60.0	90.7	74.4	81.8	64.2	46.9	60.0
LCWS	49	85.4	72.0	78.2	94.5	78.1	85.6	71.3	13.1	22.2
SCM	7	87.6	86.4	87.0	83.0	89.7	86.2	79.9	61.7	69.6

Table 2. Comparison results

1. The **MSWS** system is one of the best available products. It is released by Microsoft® (as a set of Windows APIs). **MSWS** first conducts the word breaking using MM (augmented by heuristic rules for disambiguation), then conducts factoid detection and NER using rules.
2. The **LCWS** system is one of the best research systems in mainland China. It is released by Beijing Language University. The system works similarly to **MSWS**, but has a larger dictionary containing more PNs and LNs.

As mentioned above, to achieve a fair comparison, we compare the above three systems only in terms of NER precision-recall and the number of OAS errors. However, we find that due to the different annotation specifications used by these systems, it is still very difficult to compare their results automatically. For example, 北京市政府 'Beijing city government' has been segmented inconsistently as 北京市/政府 'Beijing city' + 'government' or 北京/市政府 'Beijing' + 'city government' even in the same system. Even worse, some LNs tagged in one system are tagged as ONs in another system. Therefore, we have to manually check the results. We picked 933 sentences at random containing 22,833 words (including 329 PNs, 617 LNs, and 435 ONs) for testing. We also did not differentiate LNs and ONs in evaluation. That is, we only checked the word boundaries of LNs and ONs and treated both tags exchangeable. The results are shown in Table 2. We can see that in this small test set **SCM** achieves the best overall performance of NER and the best performance of resolving OAS.

8 Conclusion

The contributions of this paper are three-fold. First, we formulate the Chinese word segmentation problem as a set of correlated problems, which are better solved simultaneously, including word breaking, morphological analysis, factoid detection and NER. Second, we present a unified approach to these problems using the improved source-channel models. The models provide a simple statistical framework to incorporate a wide variety of linguistic knowledge and statistical models in a unified way. Third, we evaluate the system's performance on an annotated test set, showing very promising results. We also compare our system with several state-of-the-art systems, taking into account the fact that the definition of Chinese words varies from system to system. Given the comparison results, we can say with confidence that our system achieves at least the performance of state-of-the-art word segmentation systems.

References

Cheng, Kowk-Shing, Gilbert H. Yong and Kam-Fai Wong. 1999. A study on word-based and integral-bit Chinese text compression algorithms. *JASIS*, 50(3): 218-228.

Chien, Lee-Feng. 1997. PAT-tree-based keyword extraction for Chinese information retrieval. In *SIGIR97*, 27-31.

Dai, Yubin, Christopher S. G. Khoo and Tech Ee Loh. 1999. A new statistical formula for Chinese word segmentation incorporating contextual information. *SIGIR99*, 82-89.

Gao, Jianfeng, Joshua Goodman, Mingjing Li and Kai-Fu Lee. 2002. Toward a unified approach to statistical language modeling for Chinese. *ACM TALIP*, 1(1): 3-33.

Lin, Ming-Yu, Tung-Hui Chiang and Keh-Yi Su. 1993. A preliminary study on unknown word problem in Chinese word segmentation. *ROCLING 6*, 119-141.

Katz, S. M. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE ASSP* 35(3):400-401.

Packard, Jerome. 2000. The morphology of Chinese: A Linguistics and Cognitive Approach. Cambridge University Press, Cambridge.

Sproat, Richard and Chilin Shih. 2002. Corpus-based methods in Chinese morphology and phonology. In: *COOLING 2002*.

Sproat, Richard, Chilin Shih, William Gale and Nancy Chang. 1996. A stochastic finite-state word-segmentation algorithm for Chinese. *Computational Linguistics.* 22(3): 377-404.

Sun, Jian, Jianfeng Gao, Lei Zhang, Ming Zhou and Chang-Ning Huang. 2002. Chinese named entity identification using class-based language model. In: *COLING 2002*.

Teahan, W. J., Yingying Wen, Rodger McNad and Ian Witten. 2000. A compression-based algorithm for Chinese word segmentation. *Computational Linguistics*, 26(3): 375-393.

Wu, Zimin and Gwyneth Tseng. 1993. Chinese text segmentation for text retrieval achievements and problems. *JASIS*, 44(9): 532-542.