Exploiting Unlabeled Text to Extract New Words of Different Semantic Transparency for Chinese Word Segmentation

Richard Tzong-Han Tsai[†*] and **Hsi-Chuan Hung**[‡]

[†]Department of Computer Science and Engineering,
Yuan Ze University, Chung-Li, Taoyuan, Taiwan
[‡]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
thtsai@saturn.yzu.edu.tw yabthung@gmail.com
[*]corresponding author

Abstract

This paper exploits unlabeled text data to improve new word identification and Chinese word segmentation performance. Our contributions are twofold. First, for new words that lack semantic transparency, such as person, location, or transliteration names, we calculate association metrics of adjacent character segments on unlabeled data and encode this information as features. Second, we construct an internal dictionary by using an initial model to extract words from both the unlabeled training and test set to maintain balanced coverage on the training and test set. In comparison to the baseline model which only uses n-gram features, our approach increases new word recall up to 6.0%. Additionally, our approaches reduce segmentation errors up to 32.3%. Our system achieves state-of-the-art performance for both the closed and open tasks of the 2006 SIGHAN bakeoff.

1 Introduction

Many Asian languages do not delimit words by spaces. Word segmentation is therefore a key step for language processing tasks in these languages. Chinese word segmentation (CWS) systems can be built by supervised learning from a labeled data set. However, labeled data sets are expensive to prepare as it involves manual annotation efforts. Therefore, exploiting unlabeled data to improve CWS performance becomes an important research goal. In addition, new word identification (NWI) is also very important because they represent the latest information, such as new product names.

This paper explores methods of extracting information from both internal and external unlabeled data to augment NWI and CWS. According to (Tseng and Chen, 2002), new words can be divided into two major categories: Words with high or low semantic transparency (ST), which describes the correlation of semantic meanings between a word and its morphemes. We designed effective strategies toward the identification of these two new word types. One is based on transductive learning and the other is based on association metrics.

2 The Model

2.1 Formulation

We convert the manually segmented words into tagged character sequences. We tag each character with either B, if it begins a word, or I, if it is inside or at the end of a word.

2.2 Conditional Random Fields

CRFs are undirected graphical models trained to maximize a conditional probability (Lafferty et al., 2001). A linear-chain CRF with parameters $\Lambda = \lambda_1, \lambda_2, \ldots$ defines a conditional probability for a state sequence $y = y_1 \ldots y_T$ given an input sequence $x = x_1 \ldots x_T$ to be

$$P_\Lambda(y|x) = \frac{1}{f_x} \exp\left(\sum_{t=1}^{T}\sum_{k} \lambda_k f_k(y_{t-1}, y_t, x, t)\right)$$

where Z_x is the normalization that makes the probability of all state sequences sum to one; $f_k(y_{t-1}, y_t, x, t)$ is often a binary-valued feature function and λ_k is its weight. The feature functions can measure any aspect of a state transition, $y_{t-1} \rightarrow y_t$, and the entire observation sequence, x, centered at the current position, t. For example, one feature function might have value 1 when y_{t-1} is the state B, y_t is the state I, and is the character "國". Large positive values for λ_k indicate a preference for such an event; large negative values make the event unlikely.

In our CRF model, each binary feature is multiplied with all states (y_t) or all state transitions ($y_{t-1}y_t$). For simplicity, we omit them in the following discussion. In addition, we use C_0 rather than x_t to denote the current character.

3 Baseline n-gram Features

Character n-gram features have proven their effectiveness in ML-based CWS (Xue and Shen, 2003). We use 4 types of unigram feature functions: C_0, C_1 (next character), C_{-1} (previous character), C_{-2} (character preceding C_{-1}). Furthermore, 6 types of bigram features are used, and are designated here as conjunctions of the previously specified unigram features, $C_{-2}C_{-1}$, $C_{-1}C_0$, C_0C_1, $C_{-3}C_{-1}$, $C_{-2}C_0$, and $C_{-1}C_1$.

4 New Word Identification

We mainly focus on improving new word identification (NWI) using unlabeled text. Words with high and low ST are discussed separately due to the disparity in their morphological characteristics. However, it is unnecessary for our system to classify words as high- or low-ST because our strategies for dealing with these two classes are employed synchronously.

4.1 High-ST words

For a high-ST word, its meaning can be easily derived from those of its morphemes. A word's semantic meaning correlates to its tendency of being affixed to longer words. This behavior can be recorded by the baseline n-gram model. When the baseline model is used to segment a sentence containing a high-ST word, since this tendency is consistent with that is recorded in the baseline model, this word tends to be successfully segmented. For example, suppose 指南車 zhi-nan-che (compass chariot) is in the training set. The baseline n-gram model will record the tendency of 指南 zhi-nan (guide) that it tends to be a prefix of a longer word. When tagging a sentence that contains another high ST word also containing 指南, such as 指南針 zhi-nan-zhen (compass), this word can be correctly identified.

Using only n-gram features may prevent some occurrences of high-ST words from being identified due to the ambiguity of neighboring n-grams. To rectify this problem, we introduce the transductive dictionary (TD) feature, which is similar to the traditional dictionary feature that indicates if a sequence of characters in a sentence matches a word w in an existing dictionary. The difference is that the TD not only comprises words in the training set, but contains words extracted from the unlabeled test set. The transductive dictionary is so named because it is generated following general concepts of transductive learning. We believe adding TD features can boost recall of high-ST words. More details on the TD are found in Section 5. The TD features that identify high-ST words are detailed in Section 6.1.

4.2 Low-ST words

On the contrary, new words lack of ST, such as transliteration names, are more likely to be missed by the baseline n-gram model, because their morphemes' morphological tendencies are not guaranteed to be consistent with those recorded by n-gram features. For instance, suppose 天平 tian-ping (libra) only appears as individual words in the training set. The baseline model cannot identify 熊天平 xiong-tian-ping (a singer's name) because 熊天平 is a low-ST word and the morphological tendency of 天平 is not consistent with the recorded one.

In English, there is a similar phenomenon called multi-word expressions (MWEs). (Choueka, 1988) regarded MWE as connected collocations: a sequence of neighboring words "whose exact meaning cannot be derived from the meaning or connotation of its components", which means that MWEs also have low ST. As some pioneers provide MWE identification methods which are based on association metrics (AM), such as likelihood ratio (Dunning, 1993).

The methods of identifying low-ST words can be divided into two: filtering and merging. The former uses AM to measure the likelihood that a candidate is actually a whole word that cannot be divided. Candidates with AMs lower than the threshold are filtered out. The latter strategy merges character segments in a bottom-up fashion. AMs are employed to suggest the next candidates for merging. Both methods suffer from two main drawbacks of AM: dependency on segment length and inability to use relational information between context and tags. In the first case, applying AMs to ranking character segment pairs, it is difficult to normalize the values calculated from pairs of character segments of various lengths. Secondly, AMs ignore the relationships among n-grams (or other contextual information) and labels, which are abundant in annotated corpora, and they only use annotation data to determine thresholds. In Section 6.2, we illustrate how encoding AMs as features can avoid the above weaknesses.

5 Balanced Transductive Dictionary

The simplest TD is composed of words in the training set and words extracted from the unlabeled test set. The main problem of such TD is the disparity in training and test set coverage. During training, since its coverage is 100%, the enabled dictionary features will be assigned very high weights while n-gram features

will be assigned low weights. During testing, when coverage is approximately 80-90%, most tags are decided by dictionary features enabled by IV words, while n-gram features have little influence. As a result, it is likely that only IV words are correctly segmented, while OOV words are over-segmented. Loosely speaking, a dictionary's coverage of the training set is linked to the degree of reliance placed by the CRF model on the corresponding dictionary features. Therefore, the dictionary should be made more balanced in order to avoid the potential problem of overfitting. Here a dictionary is said to be more balanced if its coverage of the training set approximates its coverage of the test set while maximizing the latter. Afterward we name the TD composed of words from gold training set and tagged test set and as Naïve TD (NTD) for its unbalanced coverage in training and test set.

Our TD is constructed as follows. Given *initial features*, we use the model trained on the whole training set with these features to label the test set and add all words into our TD.

The next step is to balance our TD's coverage of the training and test sets. Since coverage of the test set cannot reach 100%, the only way to achieve this goal is by slightly lowering the dictionary's coverage on the training set. We apply n-fold cross-tagging to label the training set data: Each fold that has $1/n$ of the training set is tagged by the model trained on the other $n-1$ folds with initial features. All the words identified by this cross-tagging process are then added to our TD. The difference between the NTD and our TD is that the NTD extracts words from the gold training set, but our TD extracts words from the cross-tagged training set. Finally, our TD is used to generate dictionary features to train the final model. Since the TD constructed from cross-tagging training set and tagged test set exists more balanced coverage of the training and test set, we call such a TD "balanced TD", shorted as BTD.

6 Our NWI Features

6.1 Transductive Dictionary Features

If a sequence of characters in a sentence matches a word w in an existing dictionary, it may indicate that the sequence of characters should be segmented as one word. The traditional way is to encode this information as binary word match features. To distinguish the matches with the same position and length, we propose a new word match feature that contains frequency information to replace the original binary word match feature. Since over 90% of words are four or fewer characters in length, we only consider words of one to four characters. In the following sections, we use D to denote the dictionary.

6.1.1 Word Match Features (WM)

This feature indicates if there is a sequence of neighboring characters around C_0 that match a word in D. Features of this type are identified by their positions relative to C_0 and their lengths. Word match features are defined as:

$$\begin{aligned}&\mathrm{WM}(w = C_{-pos} \dots C_{-pos+len-1}) \\ &= \begin{cases} 1 & \text{if } w \in D \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

where $len \in [1..4]$ is w's length and $pos \in [0..len]$ is C_0's zero-based relative position in w (when $pos = len$, the previous len characters form a word found in D). If C_0 is "會" and "討論會" is found in D, $\mathrm{WM}(C_{-2} \dots C_0)$ is enabled.

6.1.2 Word Match with Word Frequency (WMWF)

Given two different words that have the same position and length, WM features cannot differentiate which should have the greater weight. This could cause problems when two matched words of same length overlap. (Chen and Bai, 1998) solved this conflict by selecting the word with higher (frequency × length). We utilize this idea to reform the WM features into our WMWF features:

$$\begin{aligned}&\mathrm{WMWF}_q(w = C_{-pos} \dots C_{-pos+len-1}) \\ &= \begin{cases} 1 & \text{if } w \in D \text{ and } \log\_\mathrm{freq}(w) = q \\ 0 & \text{otherwise} \end{cases}\end{aligned}$$

where the word frequency is discretized into 10 bins in a logarithmic scale:$\log\_\mathrm{freq}(w) = \min(\lceil \log_2 w\text{'s frequency} + 1 \rceil, 10)$

thus $q[0..10]$ is the discretized log frequency of w. In this formulation, matching words with higher log frequencies are more likely to be the correct segmentation. Following the above example, if the frequency of "討論會" is 15, then the feature $\mathrm{WMWF}_4(C_{-2} \dots C_0)$ is enabled.

6.1.3 Discretization v.s. Zipf's Law

Since current implementations of CRF models only allow discrete features, the word frequency must be discretized. There are two commonly used discretization methods: equal-width interval and equal-frequency interval, where the latter is shown to be more suitable for data following highly skewed distribution (Ismail and Ciesielski, 2003). The word frequency distribution is the case: Zipf's law (Zipf, 1949) states that the word frequency is inversely proportional to its rank (Adamic and Huberman, 2002):

$$f(x) \propto z^{-\alpha}$$

where $f(x)$ is x's frequency, z is its rank in the frequency table, and α is empirically found to be close to unity. Obviously this distribution is far from flat uniform. Hence the equal-frequency binning turns out to be our choice.

Ideally, we would like each bin to have equal *expected* number of values rather than following *empirical* distribution. Therefore, we attempt to discretize according to their underlying Zipfian distribution.

Adamic & Huberman (2002) shows that Zipf's law is equivalent to the power law, which describes Zipf's law in a unranked form:

$$f_X(x) \propto x^{-(1+(1/\alpha))},$$

where X is the random variable denoting the word frequency and $f_X(x)$ is its probability density function. Approximated by integration, the expected number of values in the bin $[a, b]$ can be calculated as

$$\sum_{a \le x \le b} x \cdot \Pr[X = x] \approx \int_a^b x \cdot f_X(d)dx$$
$$\propto \int_a^b x \cdot x^{-(1+(1/\alpha))}dx \approx \ln x|_a^b = \ln(b/a)$$
$$(\because \alpha \approx 1)$$

Thus each bin has equal number of values within it if and only if b/a is a constant, which is in a log scale. This shows that our strategy to discretize the WMWF and WMNF features in a log scale is not only a conventional heuristic but also has theoretical support.

6.2 Association Metric Features (AM)

In this section, we describe how to formulate the association metrics as features to avoid the weakness stated in Section 4.2. Our idea is to enumerate all possible character segment pairs before and after the segmentation point and treat their association metrics as feature values. Each possible pair corresponds to an individual feature. For computational feasibility, only pairs with total length shorter than five characters are selected. All the enumerated segment pairs are listed in the following table:

Feature	x,y	Feature	x,y
AM^{1+1}	c_{-1}, c_0	AM^{2+1}	$c_{-2}c_{-1}$, c_0
AM^{1+2}	c_{-1}, c_0c_1	AM^{2+2}	$c_{-2}c_{-1}$, c_0c_1
AM^{1+3}	c_{-1}, $c_0c_1c_2$	AM^{3+1}	$c_{-3}c_{-2}c_{-1}$, c_0

We use Dunning's method (Dunning, 1993) because it does not depend on the assumption of normality and it allows comparisons to be made between the significance of the occurrences of both rare and common phenomenon. The likelihood ratio test is applied as follows:

$$LR(x, y) = 2 \times (\text{logl}(p_1, k_1, n_1) + \text{logl}(p_2, k_2, n_2) - \text{logl}(p, k_1, n_1) - \text{logl}(p, k_2, n_2))$$

where $\text{logl}(P, K, M) = K \times \ln P + (M - K) \times \ln(1 - P)$, $k_1 = \text{freq}(x, y)$; $k_2 = f(x, \neg y) = \text{freq}(x) - k_1$; $n_1 = \text{freq}(y)$; $n_2 = N - n_1$; $p_1 = \text{p}(x|y) = k_1/n_1$; $p_2 = \text{p}(x|y) = k_2/n_2$; $p = \text{p}(x) = (k_1 + k_2)/N$;
N is the number of words in corpus.

An important property of likelihood ratio is that $-2LR$ is asymptotically x_1^2 distributed. Hence we can directly compute its p-value. We then discretize the p-value into several bins, each bin is defined by two significance levels $2^{-(q+1)}$ and 2^{-q}. Thus, our AM feature is defined as:

$$AM_q(x, y) = \begin{cases} 1 & \text{if the } p\text{-value of} \\ & LR(x, y) \in [2^{-(q+1)}, 2^{-q}] \\ 0 & \text{otherwise} \end{cases}$$

Since we have a constraint $0 \le q \le 10$, thus, the last interval is $[0, 2^{-10}]$. We can think that larger q implies higher tendency of current character to be labeled as 'I'.

7 External Dictionary Features

7.1 Word Match with Ngram Frequency (WMNF)

In addition to internal dictionaries extracted from the training and test data, external dictionaries can also be used. Unlike with internal dictionaries, the true frequency of words in external dictionaries cannot be acquired. We must treat each external dictionary word as an n-gram and calculate its frequency in the entire unsegmented (training plus test) set as follows:

$$\text{WMNF}_q(w = C_{-pos} \ldots C_{-pos+len-1})$$
$$= \begin{cases} 1 & \text{if } w \in D \text{ log\_ngram\_freq}(w) = q \\ 0 & \text{otherwise} \end{cases}$$

where the frequencies are discretized into 10 bins by the same way describing in previous section.

In this formulation, matching n-grams with higher log frequencies are more likely to represent correct segmentations.

8 Experiments and Results

8.1 Data and Evaluation Metrics

We use two datasets in SIGHAN Bakeoff 2006: one Simplified Chinese provided by Univ. of Pennsylvania (UPUC) and one Traditional Chinese provided by the City Univ. of HK (CITYU), as shown in Table 1.

Two unlabeled text data used in our experiments. For the CITYU dataset, we use part of the CIRB40 corpus[1] (134M). For the UPUC dataset, we use the Contemporary Chinese Corpus at PKU[2] (73M).

[1] `http://clqa.jpn.org/2006/04/corpus.html`
[2] `http://icl.pku.edu.cn/icl_res/`

			UPUC							CITYU						
			F	+/-	R_{oov}	+/-	R_{iv}	NC	NCRR	F	+/-	R_{oov}	+/-	R_{iv}	NC	NCRR
closed	1	N-grams	93.0	n/a	71.1	n/a	95.7	14094	n/a	96.6	n/a	78.8	n/a	97.3	9642	n/a
	2	(1) + AM (int_raw)	94.3	+1.3	76.4	+5.3	96.5	11655	+17.3	97.3	+0.7	80.3	+1.5	97.9	7890	+18.2
	3	(1) + WM, NTD(1)	93.4	+0.4	74.8	+3.7	95.4	13182	+6.5	97.0	+0.4	81.6	+2.8	97.3	8597	+10.8
	4	(1) + WMWF, NTD(1)	93.7	+0.7	75.0	+3.9	95.8	12719	+9.7	97.2	+0.6	82.0	+3.2	97.6	8029	+16.7
	5	(1) + WMWF, BTD(1)	94.0	+1.0	73.4	+2.3	96.7	12218	+13.3	97.4	+0.8	79.2	+0.4	98.3	7429	+23.0
	6	(1) + WMWF, BTD(2) + AM (int_raw)	94.5	+1.5	76.6	+5.5	96.7	11173	+20.7	97.5	+0.9	80.3	+1.5	98.2	7377	+23.5
	7	Rank 1 in Closed	93.3	n/a	70.7	n/a	96.3	n/a	n/a	97.2	n/a	78.7	n/a	98.1	n/a	n/a
open	8	(1) + AM (ext_raw)	94.3	+1.3	75.9	+4.8	96.6	11695	+17.0	97.3	+0.7	81.9	+3.1	97.9	7747	+19.7
	9	(1) + WMWF, BTD(8) + AM (ext_raw)	94.7	+1.7	77.1	+6.0	96.9	10844	+23.1	97.8	+1.2	82.2	+3.4	98.5	6531	+32.3
	10	(9) + WMNF	95.0	+2.0	78.7	+7.6	97.1	10326	+26.7	97.9	+1.3	84.0	+5.2	98.5	6117	+36.6
	11	Rank 1 in Open	94.4	n/a	76.8	n/a	96.6	n/a	n/a	97.7	n/a	84.0	n/a	98.4	n/a	n/a

Table 2: Comparison scores for UPUC and CITYU

Source	Training (Wds/Types)	Test (Wds/Types)
UPUC	509K/37K	155K/17K
CITYU	1.6M/76K	220K/23K

Table 1: An overview of corpus statistics

We use SIGHAN's evaluation script to score all segmentation results. This script provides three basic metrics: Precision (P), Recall (R), and F-Measure (F). In addition, it also provides three detailed metrics: R_{OOV} stands for the recall rate of the OOV words. R_{IV} stands for the recall rate of the IV words, and NC stands for NChanges (insertion+deletion+substitution) (Sproat and Emerson, 2003). In addition, we also compare the NChange reduction rate (NCRR) because the CWS's state-of-the art F-measure is over 90%. Here, the NCRR of any system s is calculated:

$$\text{NCRR}(s) = \frac{NChange_{baseline} - NChange_s}{NChange_{baseline}}$$

8.2 Results

Our system uses the n-gram features described in Section 3 as our baseline features, denoted as n-grams. We then sequentially add other features and show the results in Table 2. Each configuration is labeled with the features and the resources used in it. For instance, AM(int_raw) means AM features computed from the internal raw data, including the unlabeled training and test set, and WM, NTD(1) stands for WM features based on the NTD employing config.1's feature as its initial features.

Our experiments are conducted in the following order: starting from baseline model, we then gradually add AM features (config.2) and TD features (config.4 & 5) and combined them as our final setting (config.6) for the closed task. In the open task, we sequentially add AM features (config.8), TD features (config.9), which only exploit internal and unlabeled data. Finally, the last setting (config.10) employs external dictionaries besides all above features.

Association Metric At first, we compare the effects after adding AM which is computed based on the internal raw data (config.2). We can see that adding AM can significantly improve the performance on both datasets. Also, the OOV-recall is improved 5.3% and 1.5% on UPUC and CITYU respectively.

Transductive Dictionary Without lost of generality, we firstly use the WM features introduced in Section 6.1.1 to represent dictionary features which is denoted as config. 3 in Tables 3. We can see that the configuration with WM features outperforms that with N-grams (config.1). It is worth mentioning that even though N-grams achieve satisfactory OOV recall (0.788 and 0.711) in CITYU and UPUC, config. 3 achieves higher OOV recall.

Frequency Information and BTD To show the effectiveness of frequency, we compare WM with WMWF features. In Table 2, we can see that WMWF features (config.4) outperform WM features (config.3) on both datasets in terms of F-Measure and R_{IV}. In addition, switching the NTD (config.4) with BTD (config.5) can further improve R_{IV} and F-score while R_{OOV} slightly decreases. This is not surprising. In a BTD, most incorrectly segmented words appear infrequently. Unfortunately, the new words detected by the baseline model also have comparatively low frequencies. Therefore, these words will be assigned into the same several bins corresponding to infrequent words as the incorrectly segmented words and share low weights with them.

Combined Effects In config.6, we use the model with N-gram plus AM features as initial features to construct the BTD. In Table 2, we can see that the increase of R_{OOV}'s can recover the loss brought by using BTD and further raise

the F-measure to the level of the state-of-the-art open task performance.

In comparison of the baseline n-gram model, our approach reduces the errors by an significant number of 20.7% and 23.5% in the UPUC and CITYU datasets, respectively. The OOV recall of our approach increases 5.5% and 1.5% on the UPUC and CITYU datasets, respectively. Astonishingly, in the UPUC dataset, with limited information provided by training corpus and unlabeled test data, our system still outperforms the best SIGHAN open CWS system that are allowed to use unlimited external resources.

8.2.1 Using External Unlabeled Data

In config.9, we also use the ngrams plus AM as initial features to generate the BTD, but external unlabeled data are used along with internal data to calculate values of AM features. Comparing with config.6, we can see that R_{OOV}, R_{IV}, and F-score are further improved, especially R_{OOV}. Notably, this configuration can reduce NChanges by 2.4% in comparison of the best closed configuration.

8.2.2 Using External Dictionaries

To demonstrate that our approach can be expandable by installing external dictionaries, we add WMNF features based on the external dictionaries into the config.9, and denote this to be our config.10. We use the Grammatical Knowledge-Base of Contemporary Chinese (GKBCC) (Yu et al., 2003) and Chinese Electronic Dictionary for the UPUC and CITYU dataset, respectively.

As shown in Table 2, all metrics of config.10 are better than config.9, especially R_{OOV}. This is because most of the new words do not exist in external dictionaries; therefore, using external dictionaries can complement our results.

9 Conclusion

This paper presents how to exploit unlabeled data to improve both NWI and CWS performance. For new high-ST words, since they can be decomposed into semantically relevant atomic parts, they could be identified by the n-gram models. Using the property, we construct an internal dictionary by using this model to extract words from both the unlabeled training and test set to maintain balanced coverage on them, which makes the weights of the internal dictionary features more accurate. Also, frequency is initiatively considered in dictionary features and shows its effectiveness.

For low-ST words, we employ AMs, which is frequently used in English MWE extraction to enhance the baseline n-gram model. We show that this idea effectively extract much more unknown person, location, and transliteration names which are not found by the baseline model.

The experiment results demonstrate that adopting our two strategies generally beneficial to NWI and CWS on both traditional and simplified Chinese datasets. Our system achieves state-of-the-art closed task performance on SIGHAN bakeoff 2006 datasets. Under such most stringent constraints defined in the closed task, our performances are even comparable to open task performance. Moreover, with only external unlabeled data, our system also achieves state-of-the-art open task performance on SIGHAN bakeoff 2006 datasets.

References

L.A. Adamic and B.A. Huberman. 2002. Zipf's law and the internet. *Glottometrics*, 3:143–150.

K. J. Chen and M. H. Bai. 1998. Unknown word detection for chinese by a corpus-based learning method. *Computational Linguistics and Chinese Language Processing*, 3(1):27–44.

Y. Choueka. 1988. Looking for needles in a haystack or locating interesting collocation expressions in large textual databases. In *RIAO*.

T. Dunning. 1993. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):65–74.

Michael K. Ismail and Vic Ciesielski. 2003. An empirical investigation of the impact of discretization on common data distributions. In *Design and Application of Hybrid Intelligent Systems*. IOS Press, Amsterdam, Netherlands.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML-01*, pages 282–289.

Richard Sproat and Thomas Emerson. 2003. The first international chinese word segmentation bakeoff. In *SIGHAN-03*.

Huihsin Tseng and Keh-Jiann Chen. 2002. Design of chinese morphological analyzer. In *SIGHAN-02*.

Nianwen Xue and Libin Shen. 2003. Chinese word segmentation as lmr tagging. In *SIGHAN-03*.

G.K. Zipf. 1949. *Human Behavior and the Principle of Least Effort*. Addison-Wesley, Cambridge, MA.