1 Introduction

That a word is an intuitive unit shared in many languages seems to be borne out with alphabetical writing systems of the world. Most alphabetical writing systems incorporate a convention to mark the boundary of a word. A word boundary is typically marked with a blank space, such as in English. It is important to note that these conventionalized sociological words (as Chao (1968) terms them) do not always coincide with linguistic words. That “the White House” is a linguistic word but conventionalized to have three written and spoken segments is a well-known fact and has led to active research of so-called “multi-word units” in computational linguistics. However, in languages with conventions to mark word boundaries, such as English, the correlation between sociological words and linguistic words is very high. Hence sociological words delimited by blanks are often simply taken as linguistic units in computational linguistics. Tokenization, or word identification, is considered to be a marginal research issue in these languages.

Chinese orthography, however, does not conventionalize word boundaries. Instead, Chinese texts are composed of a string of characters, which are monosyllabic meaning bearing units. Thus, “sociological words” in Chinese correspond more closely to morphemes than words. In Natural Language Processing (NLP), this fact makes word tokenization, better known as word segmentation in Chinese language processing, a pre-requisite. The fact that segmentation remains a hot topic in Chinese language processing after some 30 years of intensive research clearly shows that it is not trivial to model human’s ability to identify words in the absence of orthographical word boundaries. However, recent advances in computational modeling and corpus-based techniques have led to increasingly more accurate word segmentation systems. It is timely to take stock of the recent progress in this field and examine the linguistic insights that have led to this progress.
2 Characterizing the Chinese word segmentation problem

In a language like Chinese where there are no conventionalized word boundaries, it is widely accepted that there are two main problems that lead to the difficulty in automatic word segmentation. The first problem is ambiguity. That is, given a string of Chinese characters, it can be legitimately segmented into different words or word sequences depending on the context in which it appears. An example would make this point clearer. Given a simplified scenario where the possibility of combination with characters outside of current string is not considered, for instance, the two-character string “可-以” should be regarded as one word (ke3yi3 ‘can’) in (1a) and two words ke3 yi3 (‘able with’) in (1b).

(1) a 他确信，加，中两国可以成为很好的合作伙伴。

“He is convinced that our two countries, Canada and China can become a very good cooperation partner.”

b 香港特别行政区可以“中国香港”名义单独同各国，各地区保持和发展金融关系。

“The Hong Kong Special Administrative Region may individually maintain financial relations with each country and region in the name of ‘China Hong Kong’.”

Similarly, the two-character string “个-人” should be treated as one word in ge4ren2 (‘individual’) in (2a) and two words in ge5 ren2 (classifier-person’) in (2b). This is not unlike the difficulty faced in segmentation of speech data. A two-character string can only be two-way ambiguous. It is either a word of two characters or two single-character words.

(2) a 另外，普金和卡斯特罗也都彼此喜欢，两人同时也在建立起了比较密切的个人关系。

in addition, Putin and Castro also all each other like, two people at the same time also establish relatively close personal
relationship.

“In addition, Putin and Castro also like each other. The two of them also established relatively close personal relationship.”

In China, “family background” used to be sufficient in determining an individual’s fate.

Longer strings have more complex ambiguity patterns. Making the same simplifying assumption of considering only the combinatorial possibilities of characters within the current string, a three-character string “A-B-C”, where A, B, C are characters, are four-way ambiguous. It can either be segmented as a single three-word sequence “ABC”, or two two-word sequences of “AB C” or “A BC”, or a three-word sequence “A B C”.

In addition to the ambiguity problem, another problem that is often cited in the literature is the problem of so-called out-of-vocabulary or “unknown” words (Chiang et al., 1992; Wu and Jiang, 1998). The unknown word problem arises because machine-readable dictionaries cannot exhaustively list all words in the language. Although the number of Chinese characters generally remains constant, derivation, compounding, and neologism are productive processes to create new words. For instance, new words can be created through compounding, in which new words are formed through the combination of existing words, or through suoxie, in which components of existing words are extracted and combined to form new words. In addition, new names are created by combining existing characters in a very unpredictable manner. Yet another productive source of new words is the transliteration of foreign words. Unknown word detection is especially challenging in Mandarin Chinese and compounded by three characteristics of Chinese orthography. First, almost all characters in Mandarin Chinese have the dual role of representing a single mono-syllabic word and as part of a multi-syllabic word. Second, the length of words in Chinese is not fixed. Although typically, more than 80% of all word tokens are contains three syllables (i.e. characters) or less, there is in theory no upper limit to the length of a compound word. And third, words are not marked conventionalized boundaries. The three characters means that any n-gram (n-character string) in a Chinese text could be a unknown word, regardless of whether any substring already matches a known entry in a dictionary or not.
The key to accurate automatic word segmentation in Chinese lies in the successful resolution of these ambiguities and an effective method to handle out-of-vocabulary words. The different approaches to Chinese word segmentation are to a large extent defined by how they tackle these two problems.

3 Approaches to Chinese Word Segmentation

3.1 Pattern Matching Approach: Dictionary Lookup

One of the most intuitive computational approaches in Chinese word segmentation is dictionary lookup. The premise is that words are character string patterns to be recognized and the computer is very good at matching patterns. Almost all earlier proposals on Chinese segmentation include a dictionary lookup module (e.g. Chen and Liu 1992). This seems to be a rather intuitive idea: given a dictionary of words, one should be able to segment a Chinese sentence that consists of a string of characters into a sequence of words based on what is in the dictionary. The first problem that one encounters with this approach is that there is ambiguity in how a string can be segmented into words, based on what is in the dictionary. In fact, how the ambiguity problem is usually characterized is still tied to how the dictionary lookup approach works. For example, given the character string “可-以”, and assuming “可-以”, “可”, and “以” are all words in the dictionary, a word segmentation system is faced with the choice of either segmenting the character string as one word “可-以”, or as a sequence of two words “可-以”. One of the early methods in resolving this ambiguity is a greedy search routine called the maximum matching algorithm. The maximum matching algorithm attempts to find the longest string of Chinese characters starting from a given point in a sentence that matches a word in the dictionary. That means that it will always prefer a two-character word over two one-character words if both are possibilities. This tends out to be a very good bet in most cases. If a two-character string is a possible word in some dictionary, in all likelihood it forms one word rather than two separate words. In the 800K-word Chinese Treebank (Xue et al, 2005), when the two characters 可-以 are next to each other and in that order, they are one word in 879 cases and two separate words in only 4 cases. This explains the early popularity of the maximum matching algorithm. However, there are also strings like “个-人”, where there is a more balanced distribution of the one-word and two-word scenarios. In the same corpus, this string forms one word in 143 cases and two separate words in 190 cases. Since the maximum matching algorithm in the purist sense does not consider context, it will always prefer the one-word segmentation.

A more serious problem with the dictionary lookup approach is that no dictionary is complete due to the dynamic nature of a natural language lexicon. On
one hand, language lexica are always in flux as neologisms add both new forms and meanings while old forms and meanings become obsolete. On the other hand, multiplication of domains and domain specific usages also made it impractical to list all possible word forms and meanings in a general dictionary. Hence most Chinese word segmentation systems which adopt the maximum matching algorithm often incorporate additional mechanisms to deal with new words. This additional mechanism often involves the use of statistical information to identify out-of-vocabulary words.

3.2 Statistical Approach: Strength of Internal Binding

When it became clear that simple pattern matching is inadequate for Chinese word segmentation, stochastic approaches were introduced. In particular, the first approaches identify the word as a unit by measuring the strength between two characters. Any two character strings which are strongly bound to each other are treated as a word. One of the earliest statistical approaches uses mutual information to measure the strength of binding in a character string (Sproat and Shih, 1990; Huang, 1995). These MI-based studies also address the issue of whether the concept of a linguistic word can be quantitatively defined. Sproat and Shih (1990), for instance, attempt to identify all words from a corpus without referring to a dictionary. The basic idea for this approach is that given a string of characters $C_1C_2\ldots$, the pair of adjacent characters with the highest mutual information value greater than a threshold is considered to be a word. The threshold is determined empirically by examining the word segmentation result. Two-character words are proposed iteratively until there are no more pairs of adjacent characters with a mutual information value greater than this threshold. The mutual information $I$ between two variables $a$ and $b$ is computed with Equation 1. Intuitively, the mutual information $I$ is high if the probability of $a$ and $b$ co-occurring is high but the probability of them occurring individually is low. The mutual information is the highest if $a$ and $b$ always occur together if they occur at all.

$$I(a; b) = \log_2 \left( \frac{P(a, b)}{P(a)P(b)} \right)$$ (1)

Sproat and Shih (1990) used a special form of mutual information called association strength with the added constraint that $a$ and $b$, which are two characters, must occur together and in that order. The association strength is computed with Equation 2. The association strength can be estimated very reliably given a large-enough corpus.
This elegant statistical approach without the use of a dictionary as a general solution to Chinese word segmentation yields reasonable accuracy for bi-syllabic words, but require more complicated and less elegant approaches for longer words. In addition, there is no reliable \textit{a priori} way to set the threshold for word segmentation, and there is no non-arbitrary solution to rule out highly collaborative non-word bigrams. Mutual information-based metrics favor low-frequency characters that always happen together. For instance, in the Chinese Treebank the character sequence “俺爹俺娘” (“my dad my mom”) occurs only once, and this is also the only context where the individual characters have occurred. The two-character sequence “爹俺” (\textit{die3 an3} “dad my”) will be assigned some of the highest MI values from the corpus but is not a word by any stretch of the imagination. The fact that MI is used to measure the strength of bonds between two characters over a corpus also means that it has no built-in solution to inherent ambiguities where two-character strings like 个-人( ‘individual’ or ‘classifier-person’ ), which can be either a word or a two-character sequence depending on the context. However, statistical approaches can be used to augment dictionary-based approaches to achieve better segmentation results. For example Sproat et al (1996) represents a dictionary as a weighted finite-state transducer. Each dictionary entry is represented as a sequence of arcs labeled with a character and its phonemic transcription, starting from an initial state 0 and terminated by a \textit{weighted} arch labeled with an empty string $\epsilon$ and a part-of-speech tag. The weight represents the estimated cost of the word, which is its negative log probability. The probabilities of the dictionary words as well as morphologically derived words not in the dictionary are estimated from a large unlabeled corpus. Given a string of acceptable symbols (all the characters plus the empty string), there exists a function that takes this string of symbols as input and produces as output a transducer that maps all the symbols to themselves. The path that has the cheapest cost is selected as the best segmentation for this string of characters. Compared with pure statistical approaches, statistical dictionary-based approaches have the guidance of a dictionary and intuitively they should outperform pure statistical approaches. However, research on Chinese word segmentation using these early approaches predates the existence of large-scale human-annotated corpora that can be used a common test set. Most studies use their own internal evaluations, and comparison between the different approaches is difficult. Sproat et al (1996), for examples evaluates the performance of their system by comparing the agreement between a human judge and the system against the agreement between two human judges, and claims their systems performs very
close to human-level agreement.

3.3 Character Tagging and Machine Learning

A recent innovation which has become the dominant approach to Chinese word segmentation involves labeling the characters with the position in which they occur within a word. In contrast to the MI-based approach which identifies words by their strong internal bonds, the position labeling of characters allows words to be identified through the identification of boundaries of words. In the original formulation of this approach, Xue (2003) used a set of four tags in representing the position of a character within a word: LR (the character is a word by itself), LL (the character starts a word but it is not a word by itself), MM (the character is in the middle of a word), and RR (the character is on the right edge of a word but it is not a word by itself). Given this formulation, the character 产 would receive different position labels based on the position in which it occurs in a word:

| LL (Left) | 产 “to come up with” |
| LR (Word by itself) | 产 “to grow wheat” |
| MM (Middle) | 产 “assembly line” |
| RR (Right) | 产 “to produce” |

Table 1: Character position tags

Ambiguity arises when a character can potentially occur in multiple positions. In the example above, “产” can occur in all four positions. However, in a specific context, it can only occur in one position and thus receive one tag. For example, the character string 可 以 (1a) would be tagged as 可/LL 以/RR, indicating the two characters form one word, but in (1b), it will be labeled as 可/LR 以/LR, indicating they are two single-character words.

The character-tagging approach proves to be a very simple and yet powerful approach when combined with machine-learning techniques that can effectively use contextual information to resolve the ambiguities when making tagging decisions. There are several advantages in formulating Chinese word segmentation as a character tagging problem where machine learning algorithms can be applied. First, Chinese words generally have fewer than four characters. As a result, the number of positions is small. Generally machine learning algorithms deal with small tagset problems more effectively than large-tagset problems. Of course, what counts as “small” needs to be determined empirically. It is not necessarily the case that two tags are better than four tags, as both are small. In fact, Xue (2003) showed that four tags work slightly better than three position tags (character by itself, starting character, middle or ending character) or two position tags (beginning character,
character that is not the beginning character). Second, although each character
can in principle occur in all possible positions, not all characters behave this way.
A substantial number of characters are distributed in a constrained manner. For
example, “们”, the plural marker for person nouns, almost always occurs in the
word-final position. Finally, the character tagging approach has a principled way
of handling out-of-vocabulary words. Out-of-vocabulary words are not handled
differently in the character tagging approach than in-vocabulary words. Al-
though Chinese words cannot be exhaustively listed and new words are bound to
occur in naturally occurring text, the same is not true for characters. The number
of character stays fairly constant and we do not generally expect to see new charac-
ters. Since the number of characters is small, when a machine learning algorithm
labels new data, it is unlikely that there are characters that it has not seen. The
same is not true if words are the targets of tagging and classification. Although
it is true that a new word may introduce hitherto unseen character positions, such
cases are rare and infrequent occurrences in the history of neologism. For instance,
the question particle 吗 is used exclusively as a word by itself (tag LR) before the
introduction of the transliterated word 吗啡 (ma3fei1, “morphine”), which has tag
LL.

The key to the success of a machine-learning based character tagging approach
is the contextual information that is fed into the machine learning algorithm. The
contextual information, called features in the machine-learning jargon, used in the
character position classification task is typically very simple. For instance, Xue’s
(2003) algorithm includes surrounding characters (previous and next two charac-
ters, using a five-character window) and position tags of previous two characters.
For example, if the current character is “们” (men5, plural marker), it is very likely
that it will occur as a suffix in a word, thus receiving the tag RR. On the other hand,
for other characters, they might be equally likely to appear on the left, on the right
or in the middle. In those cases where it occurs within a word will depend on the
surrounding characters. For example, if the current character is “爱” (ai4 “love”),
it is likely to be tagged LL if the next character is “护” (hu4 “protect”) since “爱
护” is a word. However, if the previous character is “热” (re4, “warm”), then it is
likely to be tagged RR since “热爱” is a word. Tags of previous characters are also
informative in deciding the tag of the current (target) character. For example, if
the previous character is tagged LR or RR, it means that the current character must
start a word, and should be tagged either LL or LR.

Because of its simplicity and the high accuracy it produces when appropriate
machine learning algorithms are adopted, the use of the character tagging approach
is prevalent in recent research on Chinese word segmentation and especially in the
last several International Word Segmentation Bakeoffs (Emerson, 2005; Levow,
2006; Jin and Chen, 2008). Extensions to this approach include expanded tagsets
(Zhao et al., 2006), more expressive features (Low, Ng, and Guo, 2005) and more
powerful machine learning algorithms such as Conditional Random Fields (Tseng
et al., 2005; Zhao et al., 2006; Zhao and Kit, 2008). Zhao et al (2006) showed
that by using a set of six tags (B1: beginning character, B2: second character,
B3: the third character, M: middle character that is neither B2 nor B3, R: last
color character and O: character that is a word by itself), they were able to achieve higher
word segmentation accuracy than any other tagsets. Low et al (2005) showed that
creative use of a dictionary as features to a machine learning algorithm can improve
word segmentation accuracy. For example, when the character string “新华社北
京” is examined and the character “华” is the target character whose position the
machine learning algorithm is trying to determine, and if “新华社” is a three-
character word in some dictionary, one can derive the feature that “华” is a middle
character in a three-character word. This information is shown to be useful in
improving word segmentation accuracy.

Character tagging approach combined with CRF machine learning produces
the best Sighan bakeoff result so far, with F-scores ranging just above 97%. How-
ever, it is still highly dependent on the training data size and requires substantial
learning time. This means that the algorithm cannot be used effectively for online
word segmentation of randomly extracted texts yet.

### 3.4 Word Boundary Decision

The most recent proposal of Chinese word segmentation is to treat it as Word
Boundary Decision (WBD, (Huang et al., 2007; Huang et al., 2008)). One may
have observed from the discussion of the evolution of the proposed approaches that
the performance of segmentation improves as the new models reduce the complex-
ity of the segmentation problem. Dictionary lookup involves the identification of
tens of thousands of lexical units with varying lengths. Character-tagging involves
2-6 class classification of roughly 6000 characters. WBD proposes to reduce the
task to a binary decision on one unit. The crucial innovation is that segmentation
need not involve classification of words or characters, it only involves classification
of the intervals in-between two characters. If the decision is modeled to determine
whether an interval is a word boundary or a character boundary, it is a binary deci-
sion problem. Huang et al (2007) proposes that Chinese text can be formalized as
a sequence of characters and intervals as illustrated as follows:

\[ c_0, I_0, c_1, I_1, c_2, I_2, \ldots, c_{n-1}, I_{n-1}, c_n \]

In a segmented text, all the intervals between characters are labeled as a word
boundary or as a character boundary. Each boundary item B is flanked by charac-
ters strings as its context. It is crucial to note that whether these character strings
form words or not is irrelevant in the WBD model as the character strings are
simply contextual features contributing to the decision on whether an interval is a word boundary or not. The actual implementation of the contextual information involves selection of contextual features. Huang et al.'s (2007) experiment showed that the five feature vector yields the best results: two unigrams of the character immediately before and after the interval; as well as three bi-grams before, after, and straddling the interval. For instance, in a 4-character context \( abcd \) we collect two unigrams \( b \) and \( c \) and three bi-grams \( ab \), \( bc \), and \( cd \) and compute probability of that interval being a word boundary given that particular context. These five n-grams are stored in a vector, which is labeled as Type 0 (character boundary) or Type 1 (word boundary) depending on the actual classification of that interval in the corpus: \(<ab, b, bc, c, cd, 0>\) or \(<ab, b, bc, c, cd, 1>\). An example of encoding of “时间:三月十日” is given in Table 2.

<table>
<thead>
<tr>
<th></th>
<th>ab</th>
<th>b</th>
<th>bc</th>
<th>c</th>
<th>cd</th>
<th>Typ.</th>
<th>Inter</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.500</td>
<td>0.595</td>
<td>0.003</td>
<td>0.173</td>
<td>0.021</td>
<td>0</td>
<td>时间</td>
<td></td>
</tr>
<tr>
<td>0.983</td>
<td>0.958</td>
<td>1.000</td>
<td>0.998</td>
<td>1.000</td>
<td>1</td>
<td>间:</td>
<td></td>
</tr>
<tr>
<td>1.000</td>
<td>0.998</td>
<td>1.000</td>
<td>0.713</td>
<td>0.994</td>
<td>1</td>
<td>三</td>
<td></td>
</tr>
<tr>
<td>0.301</td>
<td>0.539</td>
<td>0.010</td>
<td>0.318</td>
<td>0.054</td>
<td>0</td>
<td>月</td>
<td></td>
</tr>
<tr>
<td>0.964</td>
<td>0.852</td>
<td>1.000</td>
<td>0.426</td>
<td>0.468</td>
<td>1</td>
<td>十日</td>
<td></td>
</tr>
<tr>
<td>0.002</td>
<td>0.245</td>
<td>0.065</td>
<td>0.490</td>
<td>0.010</td>
<td>0</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Example of encoding and labeling of interval vectors in a 4-character window abcd

Note that the random probability of 0.500 is given to a feature which has no corresponding information from corpus. For instance, in the case of preceding bigram for the interval “时间” as it is at the beginning at the sentence there is no actual character bigram.

In tests reported in Huang et al. (2008), the Academia Sinica Balanced Corpus (ASBC) is used for the derivation of the n-gram collection and training data. The CityU corpus from the SigHAN Bakeoff2 collection is used for testing. In order to verify the effect of the size of the training data, the full ASBC (17 million intervals) and a subset of it (1 million randomly selected intervals) are used for training separately. Furthermore, four different classifiers, i.e., logistic regression (LogReg), linear discriminative analysis (LDA), multi-layer perceptron (NNET), and support vector machine (SVM), were tested. The segmentation results are compared with the “gold standard” provided by the SigHAN Bakeoff2. Tables 3 and Table 4 show the training and testing accuracies of various classifiers trained with the ASBC. All classifiers tested perform as expected, with their training errors increase with the size of the training data, and the testing errors decrease with it. Table 4 clearly shows that the training data size has little effect on the testing error.
while it is above 1000. This proves that once a sufficient n-gram collection is
provided for preparation of the interval vectors, classifiers can be trained with little
input.

<table>
<thead>
<tr>
<th></th>
<th>cityu</th>
<th>ckip</th>
</tr>
</thead>
<tbody>
<tr>
<td>F-measure</td>
<td>0.920</td>
<td>0.925</td>
</tr>
<tr>
<td>OOV Rate</td>
<td>0.167</td>
<td>0.187</td>
</tr>
<tr>
<td>OOV Recall Rate</td>
<td>0.920</td>
<td>0.893</td>
</tr>
<tr>
<td>IV Recall Rate</td>
<td>0.920</td>
<td>0.930</td>
</tr>
</tbody>
</table>

Table 3: Bakeoff 3: traditional Chinese (Combined corpora for training)

<table>
<thead>
<tr>
<th>No of vectors</th>
<th>Logreg</th>
<th>LDA</th>
<th>NNet</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>17,577,301</td>
<td>0.9386</td>
<td>0.9326</td>
<td>0.9373</td>
<td>0.9362</td>
</tr>
<tr>
<td>1,000,000</td>
<td>0.9386</td>
<td>0.9325</td>
<td>0.9360</td>
<td>0.9359</td>
</tr>
<tr>
<td>100,000</td>
<td>0.9389</td>
<td>0.9326</td>
<td>0.9331</td>
<td>0.9369</td>
</tr>
<tr>
<td>10,000</td>
<td>0.9393</td>
<td>0.9326</td>
<td>0.9338</td>
<td>0.9364</td>
</tr>
<tr>
<td>1,000</td>
<td>0.9373</td>
<td>0.9330</td>
<td>0.9334</td>
<td>0.9366</td>
</tr>
<tr>
<td>100</td>
<td>0.9106</td>
<td>0.9355</td>
<td>0.9198</td>
<td>0.9386</td>
</tr>
</tbody>
</table>

Table 4: Performance during test: Corpus data from SigHan Bakeoff2

Although the WBD approach has not been tested against other approaches in
any official competition, its result is remarkable in several aspects. First, it is very
fast. Running on a PC without optimizing, a bakeoff task can be completed within
50 seconds. Second, as shown in the table above, although the training corpus con-
tains nearly 20 million vectors, only 1,000 training vectors are needed to optimize
the result.

The WBD approach to Chinese word segmentation addresses two challenges
in word segmentation. First, by breaking away from established methods and not
referring to any word list, it shows that strict modularity can be maintained and
yield promising results in the basic language processing task of word identification.
Second, by being as effective with out-of-vocabulary words and by being able to
segment texts with reasonably short processing time, it suggests that robust word
segmentation for real applications in Chinese is within reach.

4 HLT requirements

For Human Language Technology (HLT) applications, it is important that the word
segmentation is done with high accuracy. Since word segmentation is considered
to be a basic technology that high-level natural language technologies such as syntactic parsing and semantic analysis depends on, mistakes in word segmentation will be propagated to high-level processing. Word segmentation accuracy is generally evaluated in terms of precision, recall and a balanced f-score. Precision is the number of words that are segmented correctly divided by the total number of words that a word segmentation system identifies as words in a corpus. Recall is the total number of correctly segmented words divided by the total number of words in the gold standard corpus that the word segmentation system is compared against. To count as a high-accuracy word segmentation system, the system has to produce word segmentation with both high precision and high recall. Generally these two scores are combined into one quantified measure, the F1-score, based on the formula below:

\[
F1 = \frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}
\] (3)

Robustness is also an important criterion for high quality word segmentation that currently receives relatively little attention. It is often the case that a word segmentation system achieves very high accuracy when evaluated on one corpus but not with another. Word segmentation accuracy often does not carry across genres. Current state-of-the-art systems have reached very high accuracy when evaluated against manually annotated gold standard corpora. For example, in the SIGHAN International Word Segmentation Bakeoffs, 95% or higher accuracy is routinely reported for high performing systems. But when used in realistic applications such as Machine Translation, there is still anecdotal evidence that current word segmentation systems are still inadequate especially when the test data is very different from the data that the word segmentation systems are trained on. This is one area of Chinese word segmentation that needs urgent improvement.

Another area that has received little attention so far is the efficiency of segmentation. The state-of-the-arts segmentation algorithms typically require large scale training and computing resources. For instance, the Microsoft Asia team (Zhao et al., 2006) has used a giga-word scale corpus and significant computing time using the CRF machine learning algorithm. It is possible to argue that computing time and storage issues can eventually be overcome with hardware development. But the requirement of large training corpus may be a crucial weakness, especially for segmentation in new domains and for online information extraction from multiple or unknown domains, exactly the context where segmentation technology is the most critical in Chinese language technology applications. The fact that new approaches to Chinese segmentation are still being proposed and published suggests that none of the current models are attested satisfactorily for HLT applications. For instance, in the most recent issue of Journal of Chinese Information Process-
ing, no less than three new models are proposed: WBD with online learning (Li and Huang, 2010), Maximum Margin Markov Network (Li and Chang, 2010), and Normalized Accessor Variety (He et al., 2010). The competitiveness of new models is a sure sign that new and more comprehensive evaluation methods, incorporating robustness and efficiency criteria are urgently needed to advance the state of art of the Chinese word segmentation technology. An evaluation criteria combining segmentation with specific HLT applications could be an attractive alternative.

5 Conclusion

We have examined the evolution of computational approaches to Chinese word segmentation. We show that much of this progress is driven by new formulations of the word segmentation problem based on new (or better use of old) linguistic insights, coupled with advances in statistical machine learning techniques. The new approaches are made possible by the availability of large quantities of human annotated and un-annotated data, thanks in large part to the existence of the Internet, which makes data collection increasingly easier. Much work remains to be done to make automatic word segmentation more robust and efficient to promote its use in real human language applications.

References


