A Web-based Approach To Chinese Word Segmentation

William J. Beksi
Department of Computer Science and Engineering
University of Minnesota
Minneapolis, MN 55455, USA
beksi@cs.umn.edu

***Abstract*—Chinese text processing requires the detection of word boundaries. This is a non-trivial step because Chinese does not contain explicit whitespace between words. Existing word segmentation techniques make use of precompiled dictionaries and treebanks. The creation of dictionaries and treebanks is a labor-intensive process and consequently they are updated infrequently. Furthermore, due to their static nature, dictionaries and treebanks lack the latest words that enter the lexicon. This paper proposes a way to leverage content on the Internet to build a bootstrapping Chinese word segmenter. The segmenter can perform automatic updates allowing it to incorporate the latest lexicon.**

***Keywords*-text mining; parsing; corpus linguistics; word segmentation; out-of-vocabulary words; automatic language processing**

I. Introduction

Word segmentation is the first step in Chinese text processing in order to perform high-level natural language processing (NLP) tasks (machine translation, named entity recognition, part-of-speech tagging, etc.). For languages such as English, word boundaries are given by whitespace or punctuation. However, Chinese does not use whitespace to delimit words. Therefore, it is necessary to segment Chinese text to represent word boundary information. This is a difficult task due to the fact that there is no unifying segmentation standard and has drawn a large body of research in the Chinese language processing community.

In Chinese text, sentences are composed of Chinese characters or *hanzi*. Most *hanzi* can occur in different positions within different words. Table I shows how the Chinese character 学 ("learn") can in occur four different positions. Since a *hanzi* can occur within different positions of a given word, word boundaries cannot be relied upon by character position.

Table I
Character occurrence in different word positions

Position	Example
Word by itself	学 'to learn'
Left	学生 'student'
Middle	化学家 'chemist'
Right	三角学 'trigonometry'

Further ambiguities in word boundaries are described in detail in [1]. For example, 文 can occur in both the first and last position of a word. It occurs in the first position in 文章 ("article") but in the last position in 中文 ("Chinese"). Consider a sentence that contains the string "中文章". A word segmenter would be faced with the problem of splitting the string between 中 and 文 thereby grouping 文章 as a word, or splitting between 文章 grouping 中文 as a word. The situation that occurs with 章 is similar to 文 since it can also occur as the first or last character of a two character word. As another example, consider the two possible segmentations of the sentence below:

1. Segmentation 1
   日文 章鱼 怎么 说?
   Japanese octopus how say
   "How do you say octopus in Japanese?"

2. Segmentation 2
   日 文章 鱼 怎么 说?
   Japan article fish how say

The first segmentation is correct whereas the second is not. This ambiguity occurs because in some contexts a *hanzi* should be considered a word by itself, but in other contexts it should be a component of a multi-character word. A human segmenting the above example can rely on knowledge and experience to resolve this ambiguity. However, a machine segmenter can only use words in the known lexicon or prior training to make the distinction.

II. Problem Description

In addition to the problem of knowing where to insert word boundaries, another outstanding problem is out-of-vocabulary (OOV) words [2]. The OOV word problem occurs due to the fact that no dictionary or treebank can possibly list all words encountered during an NLP task. There are several mechanisms for creating new words in Chinese. First, new words can be created through the concatenation of existing words, i.e. compounding. Second, a new word can be formed by using existing characters in a new combination. The third method is by transliteration which is used extensively in translating foreign names to Chinese. To successfully detect word boundaries when processing Chinese text we need to resolve the above ambiguities and properly handle out-of-vocabulary words.

The Internet provides a wealth of human generated content. In the last decade, China has seen an exponential growth in its number of Internet users. The country currently has over 513 million Internet users, nearly one

quarter of the world users [3]. New content is being generated to cater to these users (news, social media sites, etc.), and similarly the users themselves are generating content (web logs, wikis, online forums, etc.). Neologisms can often be found within this online content. This rich environment of natural language data has made corpus building an attractive idea.

Human generated content on the Internet can be found embedded within the tags of HTML pages. We leverage this generated content to construct a bootstrapping word segmenter. Initially, the segmenter begins with zero knowledge running in a lexicon building stage. Once a sufficient lexicon is constructed, it can then begin to operate as a functioning word segmenter employing a maximum matching technique. The framework we propose in this paper excels in the area of OOV word detection, a major hurdle in Chinese word segmentation research.

III. Related Work

Over the past two decades many techniques, both dictionary and statistical based, have been applied to the Chinese word segmentation problem. Depending on the application, trade-offs between runtime and accuracy tend to favor one approach over another. Techniques for performing word segmentation typically rely on a well constructed corpus for lexical development. Word segmentation using only corpus content has been researched by [4], [5], and [6].

There has been ongoing interest for many years in the computational linguistics and lexicography communities in using the Internet as a source of natural language data. Exploring the Web as a constantly growing and renewable corpus is introduced in [7]. The idea of the Web as a corpus is established and the possibilities and constraints of tapping into the Web for linguistic data are discussed.

Constructing a large corpus of Chinese text from the Internet is a challenging and difficult task. Experience in web crawling for building a large Chinese corpus is detailed in [8]. The goal of the authors was to extract out structured Chinese content from the Internet, for human judgement, before inclusion in a lexicon. We build upon this idea, removing the human component, in order to create a self-updating Chinese word segmenter.

IV. Web-based Word Segmenter Framework

HTML is the dominant markup language for web pages. The language is composed tags, enclosed in angle brackets, where content begins with a start tag and concludes with an end tag. The tags that we are concerned with are primarily the meta and anchor tags. The meta tag provides information about the HTML page. For Chinese content, this is typically a list of comma separated keywords for the given page. Anchor tags are used to create a link to another web page. They are often accompanied by a string description of the link. Other notable tags that we make use of include the paragraph tag which contains user generated content.

The framework consists of two parts: a lexicon dictionary (hash table) constructor and word segmenter. Initially, a lexicon consisting of at least a few thousand words needs to be generated in order to be able to begin word segmenting. Once the lexicon dictionary is sufficiently populated, we can then begin to segment Chinese text based on a maximum matching algorithm. The segmented text is outputted to the user.

Lexicon building is done through a feed of HTML pages. A web crawler capable of processing millions of URIs (Uniform Resource Identifiers) is a necessary prerequisite for generating a suitable lexicon dictionary. The web crawler we use to perform these duties is Heritrix [9]. Heritrix is an open source crawler developed by the Internet Archive.[1] The crawler is modular, multi-threaded, and capable of handling large crawls.

The lexicon builder consists of an HTML parser that can tokenize a string of Chinese text based on suitable heuristics. Lexicon building is persistent as tokenized words are saved to the lexicon dictionary. The maximum token size is capped at a length of 16. This size is chosen to bound the word entry lengths thus making the lexicon dictionary manageable as far as access time and memory are concerned. The lexicon dictionary is written to disk when the program exits and is available upon start up the next time the lexicon builder is instantiated. Statistics included in the lexicon dictionary are the word frequencies of each entry. When a new word is added to the lexicon dictionary its frequency is set to one. Additional occurrences of the word bump the frequency count for that word entry. Keeping track of the word frequency allows us to use frequency based heuristics when segmenting words. It also allows for a heuristic based mechanism for removing low frequency words, which could indicate false positives, from the lexicon dictionary.

An up-to-date lexicon dictionary is maintained by performing scheduled crawls of Chinese web pages. By periodically updating the lexicon dictionary we allow for the capture of OOV words as well as the maintenance of the frequency statistics.

A. Lexicon Building Heuristics

The first heuristic used during the lexicon building stage is string length. About 69% of Chinese text consists of two character words, and of the remainder, all but 1% consists of one character words [10]. This implies that we can handle about 99% of the words encountered if we always accept one and two character occurrences between HTML tags as a word.

The second heuristic involves punctuation marks. Within an HTML meta tag is a list of keywords for the given web page. We take each string of characters separated by a comma (,) as a potential word. Similarly, characters that appear between a left (‘(’) and right (‘)’) parenthesis, a left ([) and right (]) bracket, and single (’) or double (”) quotation marks are accepted as words if their length is within the bounds of a set threshold. The detection of whitespace separating a character string is also used as an indicator of a potential word.

[1] http://crawler.archive.org/

B. Maximum Matching Algorithm

For performing word segmentation we employ a backward maximum matching algorithm. The maximum matching algorithm is a greedy search routine that steps through a string of text trying to find the longest string of *hanzi* starting from a given point in the string that matches a word entry in our lexicon dictionary. For example, assume 美 ("beautiful"), 国 ("country"), and 美国 ("America") are all in the lexicon dictionary. Then the maximum matching algorithm will always favor 美国 as a word, versus 美 and 国 as two separate character words. This is because 美国 is a longer string than 美 and both of them are in the lexicon dictionary. When the segmenter finds 美 it will continue to search the string in order to determine if there is a possible extension. When it finds the word 美国 in the lexicon dictionary it will decide against inserting a whitespace between 美 and 国. Searching of the string stops and a whitespace is inserted when the algorithm can no longer extend the string of *hanzi*. This process is repeated from the next *hanzi* in the string until the end of the string is reached. The algorithm is successful since, in many of the cases, the longest string also happens to be the correct segmentation. Consider the example in (1), the algorithm will correctly determine that (1), not (2), is the correct segmentation for the string assuming 日, 日文, 文章, 章鱼, and 鱼 are all in the lexicon dictionary. The success of the maximum matching algorithm is largely determined by the completeness of the lexicon dictionary. Thus, it is important to have an up-to-date dictionary containing a robust lexicon.

C. Word Matching Heuristics

When a word hit is made using the lexicon dictionary we check the frequency of the word. If the frequency is greater than a given threshold, then we take the word as a match. The assumption is that frequently used words tend to occur more often in the embedded content of a web page, especially new terms that have recently entered the lexicon.

The capability to remove words from the lexicon dictionary also exists. At scheduled intervals we automatically prune words from the lexicon dictionary that fall below a certain frequency threshold. In addition to having a mechanism to control the size of the database, removal of low frequency words helps reduce the occurrence of false positives during word segmentation.

V. Experimental Results

Experiments were conducted using the publicly available Second International Chinese Word Segmentation Bakeoff data [11]. The Bakeoff provides training, test, and gold-standard data along with a script used to score the results. The data is provided by the Academia Sinica (AS), City University of Hong Kong (HK), Microsoft Research (MR), and Peking University (PU). The segmented results are compared to the gold-standard data, the segmented text as defined by the creator of the corpus. The data sets are composed of either simplified or traditional Chinese characters.

A. Lexicon Dictionary Construction

The lexicon dictionary was first primed using static HTML dumps provided by the Wikimedia Foundation.[2] We used the 'zh' dumps which have a compressed size of 626 MB. In addition to Chinese, dumps exist for many other languages. After processing the HTML dumps we then started web crawling with Heritrix.

A set of 1500 seeds (Chinese URIs) were randomly selected from the Open Directory Project[3] database. These seeds were then used in the Heritrix job specification to start a crawl. We configured the crawl job to only include HTML documents, excluding other document types such PDF files, image files, etc. Downloaded files are stored by Heritrix in an archival (ARC) file.

The crawl was performed for 10 consecutive days and accumulated over 40 GB of archival data. At daily intervals we scheduled a post-processing of newly closed ARC files in order to update the lexicon dictionary. The first post-processing step consists of removing HTML content from the ARC file. Next, the HTML content is feed to the lexicon builder. The lexicon builder parses the data and strings of potential Chinese words are tokenized. The tokens are added to the lexicon dictionary and the frequency statistics are updated.

Halfway through and after the crawling process was stopped, we pruned words from the lexicon dictionary that fell below a frequency threshold. This was done to help remove false positives and keep the size of the lexicon dictionary workable.

B. Segmentation Evaluation

The segmentation results are evaluated using recall and precision measures. Recall and precision are respectively defined as:

$$R = \frac{c}{N}, \quad P = \frac{c}{n}$$

In the data set, c is the number of correctly segmented words, N is the number of unique correct words, and n is the number of segmented words. The recall and precision values are then used to calculate an F-measure [12]. The F-measure is defined as:

$$F = \frac{2PR}{P+R}$$

A baseline was computed for each of the corpora using only the words in the training portion of the corpus, table II. In the table, we list the test corpus name (C), the word count (WC) for the corpus, recall (R), precision (P), F-measure (F), the out-of-vocabulary (OOV) rate, the recall on OOV words (R_{oov}), and the recall on in-vocabulary words (R_{iv}). The OOV is defined as the set of words in the test corpus not occurring in the training corpus.

After the creation of a baseline, each of the corpora were tested using the lexicon dictionary created in the data collection phase. The maximum matching algorithm turns to the lexicon dictionary when it cannot find a match in the

[2] http://dumps.wikimedia.org/

[3] http://dmoz.org/

training data. If a match is found in the lexicon dictionary and passes the frequency threshold test, then the word is chosen. The results are shown in table III.

Table II
BASELINE RESULTS USING ONLY WORDS FROM THE TRAINING DATA

C	WC	R	P	F	OOV	R_{oov}	R_{iv}
AS	122,610	0.909	0.880	0.894	0.043	0.095	0.945
HK	40,936	0.899	0.832	0.864	0.074	0.187	0.956
MR	106,873	0.941	0.899	0.920	0.026	0.040	0.966
PU	104,372	0.911	0.877	0.893	0.058	0.193	0.955

Table III
RESULTS USING THE LEXICON DICTIONARY

C	WC	R	P	F	OOV	R_{oov}	R_{iv}
AS	122,610	0.886	0.877	0.881	0.043	0.711	0.915
HK	40,936	0.875	0.846	0.860	0.074	0.723	0.911
MR	106,873	0.909	0.889	0.899	0.026	0.526	0.927
PU	104,372	0.870	0.875	0.872	0.058	0.714	0.898

VI. DISCUSSION

In constructing a lexicon dictionary based on content from the Internet we were able to increase the OOV word recall rate by 61.6% in the AS corpus, 53.6% in the HK corpus, 48.6% in the MR corpus, and 52.1% in the PU corpus. This shows that harvesting words from web pages can be a promising way to increase the OOV recall rate.

A decrease of the in-vocabulary word recall rate was seen due to the existence of false positives in the lexicon dictionary. This manifests when the maximum matching algorithm takes a longer nonword string from the lexicon dictionary as a match. We believe that an upgrade of the matching routine along with the use of linguistic knowledge based heuristics can help alleviate this problem.

We chose to initially start the construction of the lexicon dictionary with the Chinese Wikipedia dumps. During the initial lexicon building stage, we found that the dumps provide an accurate and plentiful source of lexicon strings embedded in the HTML content.

In addition to the randomly selected URI seeds, we also included URIs to popular news outlets and social media sites. The news outlets provide an important source of words that are related to current events while social media sites are a rich source of personal names and informal words and expressions.

We did not perform any deduplication of web pages in the post-processing step. This a problem that we plan to address since duplicate HTML files will artificially inflate the lexicon dictionary statistics.

VII. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a framework for building a bootstrapping Chinese word segmenter using human generated content on the Internet. The goal of this framework is to show that the OOV recall rate can be significantly boosted by using a lexicon dictionary built from Internet content. Potential applications of this framework include word segmentation tasks within a search engine while performing information retrieval. This framework is not limited to Chinese and can be applied to any language where whitespace is not used to delimit word boundaries.

The word segmenter relies on the accuracy of the content embedded in online HTML pages. As a result, it has the tendency to introduce false positives when compared to a treebank or precompiled dictionary. However, unlike a treebank bank or precompiled dictionary, it has the capability of updating itself on a daily basis and therefore a greater probability of capturing OOV words.

Future work includes the implementation of a forward-backward maximum matching algorithm or a statistical based approach to obtain better performance. We would also like to add support for GB and Big-5 character encoded web pages (only UTF-8 is currently supported). Finally, the addition of linguistic knowledge based heuristics into the segmenter would give us more precise information about the word relationship in a given context.

REFERENCES

[1] K.W. Gan, "Integrating word boundary disambiguation with sentence understanding," Ph.D. thesis, National University of Singapore, 1995.

[2] A. Wu and Z. Jiang, "Word segmentation in sentence analysis," Proceedings of the 1998 International Conference on Chinese Information Processing, Beijing, China, 1998.

[3] Internet World Stats, http://www.internetworldstats.com/, 2011.

[4] R. Sproat, C. Shih, W. Gale, and N. Chang, "A stochastic finite-state word-segmentation algorithm for Chinese," Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, New Mexico, USA, 1994.

[5] M. Sun, D. Shen, and B.K. Tsou, "Chinese word segmentation without using lexicon and hand-crafted training data," Proceedings of COLING-ACL '98, Montreal, Canada, 1998.

[6] J. Gao, M. Li, A. Wu, and C.N. Huang, "Chinese word segmentation: a pragmatic approach," Technical Report MSR-TR-2004-123, Microsoft Research, 2004.

[7] A. Kilgarriff and G. Grefenstette, "Introduction to the special issue on the web as corpus," Computational Linguistics 29(3), pages 332-347, 2003.

[8] T. Emerson and J. O'Neil, "Experience building a large corpus for Chinese lexicon construction," WaCky! Working papers on the Web as Corpus, Edited by Marco Baroni and Silvia Bernardini, 2006.

[9] G. Mohr, M. Stack, I. Ranitovic, D. Avery, and M. Kimpton, "An introduction to Heritrix, an open source archival quality web crawler," Proceedings of the 4th International Web Archiving Workshop, Bath, UK, 2004.

[10] C.Y. Suen, *Computational Studies of the Most Frequent Chinese Words and Sounds*, World Scientific, Singapore, 1986.

[11] SIGHAN, "Second International Chinese Word Segmentation Bakeoff Data," http://www.sighan.org/bakeoff2005/, 2005.

[12] C.J. Van Rijsbergen, *Information Retrieval*, Butterworths, London, 1979.