Development of a Web-Scale Chinese Word N-gram Corpus with Parts of Speech Information

Chi-Hsin Yu, Yi-jie Tang, Hsin-Hsi Chen
Department of Computer Science and Information Engineering, National Taiwan University
#1, Sec.4, Roosevelt Road, Taipei, 10617 Taiwan
E-mail: jsyu@nlg.csie.ntu.edu.tw, tangyj@nlg.csie.ntu.edu.tw; hhchen@ntu.edu.tw

Abstract

Web provides a large-scale corpus for researchers to study the language usages in real world. Developing a web-scale corpus needs not only a lot of computation resources, but also great efforts to handle the large variations in the web texts, such as character encoding in processing Chinese web texts. In this paper, we aim to develop a web-scale Chinese word N-gram corpus with parts of speech information called *NTU PN-Gram corpus* using the ClueWeb09 dataset. We focus on the character encoding and some Chinese-specific issues. The statistics about the dataset is reported. We will make the resulting corpus a public available resource to boost the Chinese language processing.

Keywords: ClueWeb09, encoding detection, part-of-speech n-grams

1. Introduction

In recent years, researchers often use web-scale texts to train their models to alleviate the data sparseness problem in statistical natural language processing. Empirical study showed that the use of web-scale resources result in more robust models (Bergsma et al., 2010). Although the web pages contain rich language usage phenomena, processing web-scale texts is not an easy task because it not only needs a lot of computation resources, but also has to filter out the noises in messy web texts. Instead of raw web pages, some search engine providers, such as Google and Microsoft, provide word N-gram corpora or Web N-gram language models for researchers (Wang et al., 2010; Liu et al., 2010; Brants & Franz, 2006). The Google N-gram corpus contains word N-grams and their counts extracted from trillion words of web pages. Microsoft N-gram language model provides an XML Web Service to users to get the probability of a word sequence.

Although the n-gram corpus and the n-gram language model are easy to be applied in an application, those resources lack of other linguistic information such as parts of speech. Therefore, researchers (Lin et al., 2010) extended an English word n-gram corpus by adding parts of speech information and developed tools to accelerate the query speed. In this paper, we aim to develop a web-scale Chinese word N-gram corpus with parts of speech information, called *NTU PN-Gram corpus*, along the similar research direction. In contrast to English, Chinese web texts are harder to process. Besides the well-known segmentation problem in Chinese, character encoding, the difference between simplified and traditional Chinese, mixing of multiple languages such as Chinese and English, and the Chinese punctuations are issues that need to be dealt with in the development of a web-scale Chinese PN-Gram (N-Grams with POS) corpus.

This paper is organized as follows. We describe the ClueWeb09 dataset in Section 2 with some statistics. In Section 3, we present an approach to resolve the issues of character encoding and mixing of multiple languages. In Section 4, we specify the segmentation tools and parts of speech tagging tool we used. In Section 5, the strategies to extract the word N-grams are illustrated. In Section 6, some statistics about the resulting dataset are shown.

2. ClueWeb09 Dataset

In 2009 CMU created ClueWeb09 dataset[1] to support information retrieval and natural language processing researches. They crawled 1,040,809,705 web pages in 10 languages. Of these, there are 177,489,357 Chinese pages, i.e., around 17.05% pages are in Chinese. The English data are encoded in UTF-8 format, but the data of the other languages are encoded in different encodings which depend on different situations. Those pages are stored in gzipped WARC format and are easy to read by using programming languages such as Java.

This dataset is huge – it has 25 TB (uncompressed) and 5 TB (compressed). Therefore, it is a good dataset for us to get the raw Chinese web pages. In this scale, processing the dataset may beyond the reach of many academic laboratories. For example, if it takes 1 second to segment and tag POS for a Chinese web page, it will take 5.6 years by a computer with a single node. In this study, we adopt a computer cluster to increase the processing speed.

3. Encodings and Mixing Languages

In Chinese, web developers use many charsets and encodings to represent their web pages. For example, in traditional Chinese, there are charsets such as Big5, CNS 11643, and Unicode. In simplified Chinese, there are charsets such as GBK, GB2312, and Unicode. Many charsets also support different encoding schemes. For example, Unicode uses UTF-8, UTF-16 and others to fit different situations, e.g., space, machine, or processing considerations. Besides, it is common to mix different languages in a Chinese text. For example, software programmers always mix Chinese and English technological terms together. An English term can be a noun or a verb. Game players may mix Japanese terms in

[1] http://lemurproject.org/clueweb09.php/index.php

Chinese documents in a game forum of Nintendo. Those issues complicated the encoding detection and language identification.

We use an algorithm to deal with encoding detection and language identification. Given a web page, we explore the possible encodings in a specific order, convert the page from the guessed encoding into Unicode, and detect if the converted text is a valid Chinese text. The first encoding passing the test is regarded as the encoding of the web page. The details are described in Algorithm 1.

Algorithm 1. Encoding detection and language identification

```
Input: A web page
Output: Its encoding scheme and the language it belongs
1.  for each encoding E
2.      Convert the page from E to Unicode (UTF-16)
3.      Compute the percentage of U+FFFDs in the
            converted page
        if the page is a valid Chinese page
4.      then return the encoding and the related
                language
```

At step 1, we try a list of encodings in a specific order. The order is determined by the web page itself and a global list of encodings. Li & Momoi (2001) proposed a Mozilla Character Detector (Chardet) based on the content of a web page. They tested their approach in 100 popular web sites, and resulted in 100% detection accuracy. Although we found the detection of Chardet is not perfect in large scale document set like ClueWeb09, it is still a good tool to start. We first adopt their toolkit to detect the possible encoding. If the encoding fails at step 4, then we explore HTTP header encoding and encoding information in HTML metadata of the given web page. If these explorations fail at step 4 again, we try a predefined list of encodings. The order of encodings is determined by the encoding frequency of web pages detected by Mozilla Chardet. Table 1 lists the first eight of the distribution of encodings in a reference corpus. Of course, the statistics just gives a reference because Mozilla Chardet may make wrong detection. From this Table, GB2312 is explored first, then UTF-8, Big5, GB18030, and so on.

Encoding	# Pages	Encoding	# Pages
GB2312	105,200,146	GBK	250,796
UTF-8	35,528,195	Windows-1252	220,917
Big5	17,795,786	ISO-8859-1	141,732
GB18030	14,469,984	Windows-1256	22,039

Table 1: The frequency detected by Mozilla Chardet

At step 2, we adopt Java to convert the given page from the guessed encoding E to UTF-16. During conversion, the system uses U+FFFD to replace those characters that cannot have valid mappings. The 'Replacement' character U+FFFD is useful at step 3 to compute the ratio of (U+FFFD)s to the total tokens in the converted page. Note that Roman alphabets are much easier to be transformed, thus they are not counted in the total tokens. At step 4, we use a threshold to determine if the conversion is successful.

Simplified Chinese, Traditional Chinese, Japanese, and Korean (CJK) characters have their own code points in Unicode, which are grouped in consecutive range. We count the ratio of the characters in the given web page to determine its major language at step 4 and to filter out the non-Chinese documents. Finally, the total number of valid Chinese pages used in this study is 173,741,587.

4. Segmentation and POS Tagging

After a valid Chinese web page is determined, we use Jericho HTML Parser[2] to extract the pure text in RFC3676 format from this page. Next we use Stanford Chinese Word Segmenter (Tseng et al., 2005) and Stanford Log-linear Part-Of-Speech Tagger (Toutanova et al., 2003) to process the text. The tagger has been demonstrated to have the accuracy 94.13% on a combination of simplified Chinese and Hong Kong texts and 78.92% on unknown words. The POS tag set adopted is LDC Chinese Treebank POS tag set. For those traditional Chinese texts, we translate them into simplified Chinese by using character-based translation. The segmentation and the tagging are the most time consuming steps in the development procedure.

5. Extracting N-grams with POS

Not all sentences in a web page are useful for N-gram extraction. We propose two strategies to select sentences. First, a sentence is identified by using a full stop, an exclamation mark, or a question mark in full-width, half-width, or ideo-graph character forms. Next, we ignore those sentences which contain less than 3 words or 5 characters. Such short sentences are usually menu items or button names in a web page. A sentence start marker <S> and a sentence end marker </S> are further inserted into the beginning and the ending positions, respectively.

We transform some types of terms in a sentence into specific class names. Those numbers that are smaller than 11 digits are replaced with zeros. The English words are transformed to foreign words <FW>. The words containing only digits and alphabets are replaced with a class <CHSQ>. We consider those words that occur more than $count_1$ times as unigram tokens. The remaining words are converted into a class <UNK>.

All n-gram patterns (n in {2, 3, 4, 5}) that occur more than $count_n$ times are extracted and placed into the final corpus. We adopt the representation scheme similar to (Lin et al., 2010). The following shows an example.

有效 5991459 JJ | 10377337 VA | 4295950 AD | …

It indicates that term 有效 (effective) occurs 5,991,459 times as JJ (other noun-modifier) and 10,377,337 times as VA (predicative adjective), and so on.

Now we have to determine the occurrence threshold, $count_n$ (n in {1, 2, 3, 4, 5}), for each n-gram. The minimum occurrence count of unigrams in Google

[2] http://jericho.htmlparser.net/docs/index.html

Chinese Web N-gram corpus is 200. In this paper, we adopt the same threshold, i.e., set *count*$_1$ to 200. It means a word will be regarded as a unigram token, when it occurs more than 200 times.

For bigram or higher, the threshold is comparatively harder to determine. If the occurrences of each POS sequence are 40 on average, and a bigram has 10 different POS tags, then the *count*$_2$ must be set to 400. That will result in much fewer n-gram patterns. On the other hand, if *count*$_2$ is set to 40, then the occurrence count of each POS is 4 on average when the bigram has 10 POS tags. This is not a reliable count to estimate the occurrences of POS tags. Considering the issue that the possible POS tags may vary largely across different bi-grams, it is hard to ensure that each POS has a minimum occurrence. Therefore, we leave the issue of finding reliable count of POS to corpus users, and consider the occurrence counts of n-grams only. We believe that corpus users will make the best choice to fit their applications.

The accumulated number of bi-grams is shown in Figure 1. If the minimum frequency is 1, it will contain all bi-grams. We set *count*$_2$ to 40 (the same as Google N-grams) to get a reliable n-gram count. Other *count*$_n$ are also set to 40 for the same reason.

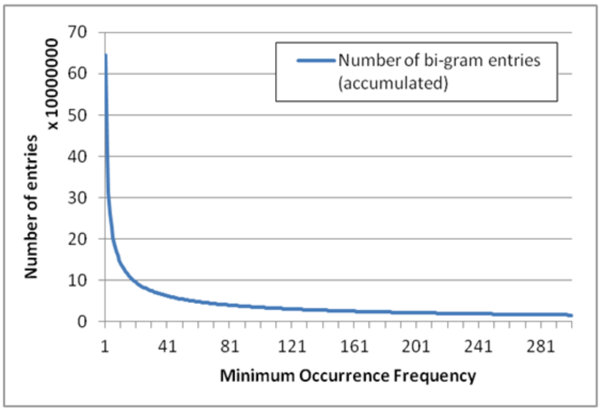

Figure 1: The accumulated number of bi-gram entries

Table 2 further shows the statistics of the word N-grams. The numbers of unique entries are listed in the 2nd column for each N. The number of unique entries whose frequencies are equal to 1 is listed in the 3rd column. The last column shows the number of unique entries of frequencies larger than 2,000.

N	#unique entries	#unique entries (freq.=1)	# unique entries (freq.>2,000)
1	107,902,213	48,640,381	453,949
2	645,952,974	238,765,583	3,705,672
3	4,184,637,707	1,873,354,994	5,734,370
4	10,923,797,159	5,505,549,988	4,724,136
5	17,098,062,929	9,381,561,487	3,274,034

Table 2: Statistics of the word N-grams

6. Statistics of NTU PN-Gram Corpus

The statistics of the resulting dataset is shown in Table 3.

Number of web pages	173,741,587
Number of sentences	9,598,430,559
Number of tokens (terms)	141,179,769,123
Number of digit terms	4,308,254,253
Number of foreign words	4,095,774,930
Number of character sequences	29,078,949,574
Average sentences per page	55.2
Average tokens per sentence	14.7

Table 3: The statistics of the resulting dataset

Compared with Google Chinese Web N-grams corpus, which contains 882,996,532,572 tokens and 102,048,435,515 sentences, the size of our dataset, i.e., 141,179,769,123 tokens and 9,598,430,559 sentences, is comparatively smaller. The statistics of the *NTU PN-Gram corpus* is shown in Table 4.

N	# NTU PN-Grams	# Google Chinese N-Grams
1	2,219,170	876,004[3]
2	62,728,971	281,107,315
3	200,066,527	1,024,642,142
4	294,016,661	1,348,990,533
5	274,863,248	1,256,043,325

Table 4: Comparison of *NTU PN-Gram corpus* and Google Chinese Web N-gram corpus

Total 48.59% of Google Chinese unigrams can be found in our unigrams. The number of unigrams we extracted is much larger than that of Google dataset. One possible reason is that the segmentation tools adopted in these two datasets are different. Google Chinese Web N-gram corpus uses a fixed vocabulary set with frequency and finds the highest frequency product from all possible segmentations. In contrast, we use CRF classifier to segment sentences. The length distribution of the words in *NTU PN-Gram corpus* and Google Chinese Web N-gram corpus is specified in Table 5.

#char	NTU PN-Grams	Google Chinese unigrams
1	16,186	21,517
2	1,100,467	467,047
3	891,401	303,913
4	156,542	66,160
5	35,739	12,961
6	11,859	4,406
7+	7,420	0

Table 5: Distribution of word length in unigrams

There are more words with one Chinese character (21,517) in Google Chinese Web N-grams corpus than ours. Besides, the maximum word length in Google Chinese is 6, and is 30 in ours. We adopt longer word length limitation to preserve the huge variations in web pages such as translated foreign names and special usages in blog posts. The following show some examples.

[3] This value did not contain 740,146 non-Chinese tokens in Google unigrams.

李奥纳多狄卡皮欧	741	// Leonardo DiCaprio
格温妮丝・帕特洛	1854	// Gwyneth Paltrow
灌水水水水水水水	1721	// add waaaaater
爽爽爽爽爽爽爽爽爽	975	// haaaaaappy
讚讚讚讚讚讚讚讚讚	694	// gooooooooood
哈哈哈哈哈哈哈哈	7297	// ha ha ha …

Because we segment the web pages directly without text pre-filtering, there are a lot of segmentation errors. Many cases, e.g., encoding detection error, special formats of named entities, and so on, may result in the segmentation errors. The following show some error examples.

侈媸汉绀稠椒尴艚	2720	// encoding error
2009 一球成名	1693	// year + set phrase
三 2008151	2077	// week + date
五 9:00-11:30	518	// week + time
周六 9:00-17:30	28683	// week + time

The above examples suggest that we need a text filtering approach similar to Gao et al. (2002) to get a more clean result. Furthermore, a named entity recognition module is preferred to improve the segmentation performance. Nevertheless, this kind of data is valuable for researchers to study the segmentation problems in handling the texts on the web.

Table 6 shows the distribution of POS tags in this corpus. The most frequent tag is common noun (NN). Because most of English words are tagged as NN, the frequency of foreign words (FW) is incredibly small comparing to the 4,095,774,930 foreign words we detected. This result shows that the mixing of foreign words with Chinese words is very common in web texts. The Chinese POS taggers need pay more attentions on handling this issue.

POS	Count	POS	Count
NN	51,834,540,030	NT	916,283,057
VV	20,322,435,225	VE	690,891,420
PU	17,340,163,283	SP	388,915,016
AD	8,185,400,528	OD	338,071,176
NR	7,231,680,516	MSP	226,901,336
CD	7,086,334,344	ETC	184,901,739
M	4,106,129,466	CS	181,125,510
JJ	3,356,058,063	BA	157,676,679
P	2,714,469,810	DEV	150,971,266
DEG	2,583,261,290	SB	104,041,348
PN	2,547,858,264	DER	59,236,647
DEC	2,115,018,767	FW	48,259,506
VA	1,709,176,978	LB	34,506,912
DT	1,560,629,878	IJ	10,812,047
LC	1,532,407,434	EOS	488,418
CC	1,324,772,696	X	170,990
VC	1,164,454,992	ON	23,137
AS	971,701,355	N/A	N/A

Table 6: Distribution of POS tags

7. Conclusion

We develop a Chinese word N-gram corpus with parts of speech information in this paper. Some important issues in preparing such a corpus are addressed and discussed. In the future, we will provide a tool similar to (Lin et al., 2010) to speed up searching the PN-Gram corpus. Besides, we plan to apply this corpus to NLP applications such as sentiment orientation detection. We hope that this public available corpus can boost the performance of NLP applications.

8. Acknowledgements

Research of this paper was partially supported by National Science Council (Taiwan) under the contracts 98-2221-E-002-175-MY3 and 99-2221-E-002-167-MY3, and Excellent Research Project of National Taiwan University. We are grateful to Computer and Information Networking Centre, National Taiwan University for the support of high-performance computing facilities.

9. References

Bergsma, S., Pitler, E. & Lin, D., 2010. Creating robust supervised classifiers via web-scale N-gram data. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Uppsala, Sweden: Association for Computational Linguistics, pp. 865–874.

Brants, T. & Franz, A., 2006. Web 1T 5-gram Version 1.

Gao, J. et al., 2002. Toward a unified approach to statistical language modeling for Chinese. *Journal ACM Transactions on Asian Language Information Processing (TALIP)*, 1(1), pp.3–33.

Li, S. & Momoi, K., 2001. A composite approach to language/encoding detection. *19th International Unicode Conference* (San Jose).

Lin, D. et al., 2010. New Tools for Web-Scale N-grams. In N. Calzolari et al., eds. In *Proceedings of the Seventh conference on International Language Resources and Evaluation*. Valletta, Malta: European Language Resources Association (ELRA).

Liu, F., Yang, M. & Lin, D., 2010. Chinese Web 5-gram Version 1.

Toutanova, K. et al., 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology* - Volume 1. Edmonton, Canada: Association for Computational Linguistics, pp. 173–180.

Tseng, H. et al., 2005. A Conditional Random Field Word Segmenter. In *Fourth SIGHAN Workshop on Chinese Language Processing*.

Wang, K. et al., 2010. An overview of Microsoft web N-gram corpus and applications. In *Proceedings of the NAACL HLT 2010 Demonstration Session*. Los Angeles, California: Association for Computational Linguistics, pp. 45–48.