SNS based Internet Words Extraction and Application Case Studies

Li Xiong
School of Computing Science, Simon Fraser University
Burnaby, BC, Canada V5A1S6
lxa27@sfu.ca

Abstract

SNS based Internet words extraction is a fundamental work in social network sentiment analysis and Chinese word segmentation. In this paper, we propose an information theory based approach to extract popular Internet words from the Chinese weibo corpus. Based on the Internet words extracted, we conduct case studies on the possible applications which include Chinese word segmentation and daily hot topic discovery. Our experiments on Chinese weibo data validate that the proposed method is effective and efficient in extracting Internet words and works on the applications mentioned above.

1 Introduction

Currently, most Chinese social network analysis research and applications rely on the identification of newly emerging Internet words. Popular Internet words on social network usually stand for the hot topics and mind trends of the public which are critical in sentiment analysis and opinion mining. What is more, adding Internet words into traditional Chinese word dictionary can improve the quality of Chinese word segmentation which can further improve the performance of Chinese-to-English translation.

Previous research on new words extraction include new words discovery on very large corpus(Zhang et al., 2000; Chen and Ma, 2002), approaches on fast substring counting(Yamamoto and Church, 2001), models(Berger et al., 1996) for unknown words extraction and studies on various statistic features(Church and Hanks, 1990) of new words. However, to the best of our knowledge, none of them conducts Internet words extraction on social network like weibo, and no work further study how to apply the method to extract Internet words on other applications such as daily hot topic discovery.

In this paper, we fill the gap between unknown words extraction and its application. Firstly, we propose an algorithm to extract Internet words from weibo data. Secondly, we conduct two case studies: 1) Improve Chinese word segmentation by adding Internet words into dictionary. 2) Discover daily popular topics on weibo. We evaluate the effectiveness of our Internet words extraction algorithm by comparing with a manually made Internet words list.

The rest of this paper is organized as follows: The proposed approach for Internet words extraction is explained in Section 2. Experiments are presented in Section 3. Followed by two application case studies in Section 4. Section 5 introduces some related work. Section 6 concludes this paper and suggests future work.

2 Internet Words Extraction Algorithm

Traditionally, new words are the snippets after word segmentation without matching in existing dictionary. However, we get these snippets by a Chinese word segmentation program using the very existing dictionary. In other words, the quality of word segmentation itself relies on the completeness of the dictionary, if there is no new words in the dictionary, the result of word segmentation is unreliable. Based on this observation, we propose an approach which can extract new words but not dependent on any word segmentation model, just consider the features of new words themselves.

The process of Internet words extraction by our approach is shown in Figure 1. At first, we crawl data from weibo which is the weibo text posted by users. We run data preprocessing program to clean the raw weibo data. In the data cleaning step, we remove four kinds of content in raw data: 1) Location information. 2) Repost indicator(i.e.

"//@userID"). 3) Stop words. 4) Punctuation and numbers. 5)Words with low tf-idf value. The next step is to run our dictionary free Internet words extraction algorithm. The details of proposed algorithm is explained below.

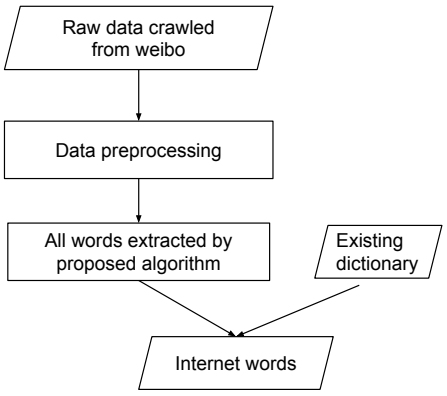

Figure 1: Internet words extraction process of proposed approach

2.1 Internet Words Extraction Algorithm

This is the third step of the whole Internet words extraction process. The input of this step is the clean data output from the second step and the output are all the meaningful words in the weibo data. The so called meaningful words stand for the words which have meanings but not nonsense word fragments.

In this algorithm, we measure the possibility that whether a fragment can be a meaningful word or not by three features: Term Frequency(TF), Mutual Information(MI) and Left/Right Entropy(LRE).

Term Frequency represents the frequency of a word fragment in the whole data set. A meaningful word usually has TF greater than one threshold value.

Mutual Information is a measure of two variables' dependency. Let $p(x)$ and $p(y)$ denote the marginal probability distribution functions of X and Y respectively. The mutual information of two random variables X and Y is:

$$I(X,Y) = \sum_{y\in Y}\sum_{x\in X} p(x,y)\log\frac{p(x,y)}{p(x)p(y)} \quad (1)$$

In our problem, X and Y stand for fragments which can be more than one characters. So we define MI of a fragment S in our problem as:

$$I(S) = \log\frac{p(S)}{\prod_{i=0}^{len(S)} p(s_i)} \quad (2)$$

In Equation 2, s_i is the i^{th} character in fragment S. $I(S)$ indicates the compactness of the character in S. If all s_i appear incidentally, then $p(S)$ should be almost the same with $\prod_{i=0}^{len(S)} p(s_i)$. So that $I(S)$ will be very small. On the contrary, if all s_i appear coincidentally, $I(S)$ will be much larger than the incidental case.

Entropy is a measure of the uncertainty of a random variable. The more certain a variable is, the smaller the entropy of the variable will be. **Left/Right Entropy** are used to measure how rich the given fragment's left and right neighborhoods. For example, given a fragment S="abacbad", the left neighborhood of letter 'a', denoted by $leftNeigh$ = {b} and its right neighborhood, denoted by $rightNeigh$ = {c,d}. The left entropy $lEntropy$ and right entropy $rEntropy$ of 'a' are defined as:

$$lEntropy(a) = -\sum_{i\in leftNeigh} p(i)\log p(i) \quad (3)$$

$$rEntropy(a) = -\sum_{i\in leftNeigh} p(i)\log p(i) \quad (4)$$

The richer the left/rigt neighborhoods of a fragment are, the higher possibility that this fragment is a meaningful word. In above example, 'a' has only one left neighbor which is 'b', so $lEntropy(a) = -1\log 1 = 0$. Because we are quite sure that the left neighbor of 'a' must be 'b'. The right entropy of 'a' is $rEntropy(a) = -(0.5\log 0.5 + 0.5\log 0.5) = 1$.

In our algorithm, $min(lEntropy, rEntropy)$ is used to denote the LRE(left/right entropy) feature of one fragment.

Till now, we have described all the features used in our algorithm to decide whether a fragment is a meaningful word or not. However, we leave a critical problem unsolved, that is how to get all the fragments? In our algorithm, we treat any substring with length less than or equal to $d(d = 5)$ in the corpus as a potential word fragment.

The algorithm is listed in Algorithm 1.

2.2 Algorithm Discussions

In the data cleaning step, we did not use the idf value computed from the whole web pages but compute idf value directly by the input corpus. The

Algorithm 1 Internet Words Extraction Algorithm

Input: Clean data after step two in Figure 1 and existing dictionary(not includes Internet words)

Output: Internet words extracted

```
for each weibo(tweet) in corpus do
    for substring s in tweet with len(s) ≤ d do
        fragments.add(s)
        leftNeighbor.add(left neighbor of s)
        rightNeighbor.add(right neighbor of s)
        frequency(s).increase
    end for
end for
Compute MI according to Eq. 2
Compute LRE according to Eq. 3, Eq. 4
for fragment f in fragments do
    if TF(f) ≥ minTF and MI(f) ≤ maxMI
    and LRE(f) ≥ minLRI then
        meaningfulWords.add(f)
    end if
end for
for word w in meaningfulWords do
    if w is not in dictionary then
        output w as Internet words
    end if
end for
```

reason is that the content of weibo data is quite different from it on the whole web pages.

Dealing with fragment with length equals one is a non-trival issue because the features of a word used in our algorithm do not work on fragment with only a single character. Fortunately, according to the study (Wang, 2011), Internet words are mostly with more than one characters. Thus, in our algorithm, we simply ignore all the fragments with length equals one.

Our algorithm scans the corpus once and uses dictionaries to store term frequency, mutual information etc. Therefore, the time complexity of our new Internet words extraction algorithm is $O(nlogn)$, n stands for the number of total characters in the corpus. The proposed algorithm is also easy to implement in parallel, because all the time-consuming parts are about counting which is the classic task of MapReduce framework.

3 Experiments

3.1 Data

We evaluate our Internet words extraction algorithm on Chinese microblog(weibo) data. Because no previous work extracted Internet words from weibo data, we do not have ground truth. We manually made a newly emerging Internet words list by referring to the wikipedia page as the ground truth. We compare the output of our algorithm with the manually made Internet words list to get the precision and recall. The dictionary we used to remove the non-Internet words is from SogouLab's web corpus dictionary (SogouW, 2006).

The statistics of data crawled from weibo are shown in Table 1. The total number of words in SogouW is 157201. The number of Internet words we manually made is 56 which covers almost all the popular Chinese Internet words used on social network like weibo recent years. The code and the dataset are available at: `https://github.com/Parallelli/InternetWordsExtraction`

Type	Number
weibo in corpus	20080
fragments in corpus	841892
characters in corpus	695937

Table 1: Statistics of data crawled from weibo

3.2 Setup

In our experiment, we set the maximum length of a fragment as $d = 5$, and the three threshold values: $minTF = 20$, $maxMI = -100.0$ and $minLRE = 100.0$. These parameters are set according to the empirical studies on the weibo corpus.

The evaluation metrics for the overall Internet words extraction effectiveness and quality are recall, precision and F-measure. $CorrInterWordsExtra$ stands for the number of correct Internet words extracted by our algorithm; $totWordsExtra$ stands for the total number of Internet words extracted by our algorithm; and $totInterWords$ stands for the total number of Internet words in our manually made list. In our experiments, we set $\beta = 1.0$, so that we actually use F_1 score.

$$Precison = \frac{CorrInterWordsExtra}{totWordsExtra} \quad (5)$$

$$Recall = \frac{CorrInterWordsExtra}{totInterWords} \quad (6)$$

$$F_\beta = (\beta^2 + 1.0)\frac{Recall \times Precision}{\beta^2 \times Precision + Recall} \quad (7)$$

Exp.	Features	Precision	Recall	F_1
1	TF	10.9%	100%	19.65
2	MI	5.25%	100%	9.98
3	LRE	11.9%	100%	21.27
4	TF,MI	34.4%	100%	51.1
5	TF,LRE	11.9%	100%	20.96
6	MI,LRE	21.71%	100%	35.67
7	TF,MI,LRE	**58.9%**	**100%**	**74.13**

Table 2: Experimental results of Internet words extracted by proposed approach

Features	#Output	#Positive
TF,MI,LRE	96	56

Table 3: Output of Exp.#7

3.3 Results

In total 56 distinct Internet words have been extracted by our approach from weibo corpus. We compute recall and precision by comparing them with manually made Internet words list. Details are listed in Table 2.

In the experiments, when test single feature, we set the threshold of TF as 20, MI as -100.0 and LRE as 100.0. Table 2 shows that the combination of features works better than single feature when extract Internet words. And the combination of TF, MI and LRE works the best.

The result in Table 3 shows that there are in total 96 Internet words extracted by our algorithm, and 56 of them are positive samples. In fact, there are in total 56 real Internet words in our experiment settings. Therefore, our algorithm gets a recall 100%. Even though the precision is not high, as there are only 96(very small) words of output, we can do human review efficiently to get the final Internet words.

4 Application Case Studies

4.1 Daily Hot Topic Discovery

The intuition behind daily hot topic discovery is that: Today's hot topic is different from previous. Based on this intuition, we designed a simple but effective algorithm to mine today's hot topics. We first crawl today and yesterday's weibo data, and run Internet words extraction algorithm stated in Section 2. We got two lists: yesterday and before yesterday's Internet words, denoted by Y, today and before today's Internet words, denoted by T. Obviously we can see that T is a superset of Y. $T \setminus Y$ is the Internet words that only occur today, namely, today's hot topics.

Table 4 shows the experimental result of daily hot topic discovery. We select the top 5 new Internet words in list T, and compare it with the hot topics given by weibo officially. We find that the 4 out of 5 hot Internet words our algorithm returned are the same with the ones given by weibo officially.

topicID	Ours	Official
1	微博上市 (“weibo IPO”)	微博上市 (“weibo IPO”)
2	舌尖 (“Bite”)	舌尖上的中国 (“A Bite of China”)
3	一道菜 (“One dish”)	一道菜证明你是哪里人 (“One dish to show where are you from”)
4	爸爸去哪 (“Where are you going, Dad?”)	爸爸去哪儿 (“Where are you going, Dad?”)
5	且珍惜 (“Cherish”)	马航飞机失联 (“MH airplane missing”)

Table 4: Experiment results of today's hot topic discovery

4.2 Chinese Word Segmentation

The performance of Chinese word segmentation is influenced by the completeness of dictionary. In this section, we will describe our trial on Chinese word segmentation with Internet words adding into dictionary.

Because there is no up-to-date work done to add newly Internet words to dictionary to segment weibo data, we do not have training data and ground truth. Our solution is to write a word segmentation using unigram. We test the quality of Chinese word segmentation by using two different dictionaries along with this unigram segmentation program. The basic dictionary is download from Anoop Sarkar's page (Sarkar, 2014). We take one Chinese sentence as an example to show the power of Internet words.

Input:
金胖子这货车很赞(“Jinpangzi's car is great”)
Output from Internet words free segmentation program:
金/胖子/这/货车/很/赞(“Fat Jin/ this / van / very / good)
Output from Internet words plugin segmentation

program:
金胖子/这货/车/很赞(“Jinpangzi / this guy/ car/ great)
We added Internet words “金胖子” and “这货” into the original dictionary, so that we segmented the sentence on weibo correctly.

5 Related Work

Generally speaking, there are two categories of algorithm to extract unknown words. There are an ocean of publications in this area so that we cannot list all of them. The one closest to our work is (Chen and Ma, 2002), they applied both statistic measures such as mutual information and morphological rules for unknown words extraction. Based on this work, (Ling et al., 2003) improved the performance of unknown words extraction by introducing POS and chucking technology into the system. However, Internet words extraction from social network is virgin. With regard to social network hot topic discovery, (Kim et al., 2013) proposed a geographic clustering analysis algorithm to detect hot topic which is based on the social topics across provinces. However, no work tries the simplest but effective way proposed in this paper to discover daily hot topics.

6 Conclusion

In this paper, we proposed an effective and efficient algorithm to extract Internet words on social network by utilizing the intrinsic features of Internet words. Our work is also the first try applying Internet words extraction method to discover daily hot topics. We also did experiments to show that Internet words are crucial for Chinese word segmentation, which can future improve Chinese-to-English translation quality.

As future work, Internet words extraction method can be extended to split regions when collaborates with location information. And personalized recommendation can also be improved through the analysis of new Internet words on social network.

Acknowledgments

Many thanks to Dr. Anoop Sarkar for his interesting and informative lectures from which I learned a lot natural language processing knowledge.

References

Adam L Berger, Vincent J Della Pietra, and Stephen A Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational linguistics*, 22(1):39–71.

Keh-Jiann Chen and Wei-Yun Ma. 2002. Unknown word extraction for chinese documents. In *Proceedings of the 19th International Conference on Computational Linguistics - Volume 1*, COLING ’02, pages 1–7, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kenneth Ward Church and Patrick Hanks. 1990. Word association norms, mutual information, and lexicography. *Computational linguistics*, 16(1):22–29.

Hwi-Gang Kim, Seongjoo Lee, and Sunghyon Kyeong. 2013. Discovering hot topics using twitter streaming data: Social topic detection and geographic clustering. In *Proceedings of the 2013 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining*, ASONAM ’13, pages 1215–1220, New York, NY, USA. ACM.

Goh Chooi Ling, Masayuki Asahara, and Yuji Matsumoto. 2003. Chinese unknown word identification using character-based tagging and chunking. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 2*, ACL ’03, pages 197–200, Stroudsburg, PA, USA. Association for Computational Linguistics.

Anoop Sarkar. 2014. zh-wseg.train.utf8. http://www.cs.sfu.ca/ãnoop/teaching/cmpt-413-spring-2014/.

SogouW. 2006. http://www.sogou.com/labs/dl/w.html.

Zhaochen Wang. 2011. Emergence of chinese internet slang.

Mikio Yamamoto and Kenneth W Church. 2001. Using suffix arrays to compute term frequency and document frequency for all substrings in a corpus. *Computational Linguistics*, 27(1):1–30.

Jian Zhang, Jianfeng Gao, and Ming Zhou. 2000. Extraction of chinese compound words: An experimental study on a very large corpus. In *Proceedings of the second workshop on Chinese language processing: held in conjunction with the 38th Annual Meeting of the Association for Computational Linguistics-Volume 12*, pages 132–139. Association for Computational Linguistics.