Domain-Specific New Words Detection in Chinese

Ao Chen[1], **Maosong Sun**[1]

[1]Department of Computer Science and Technology,
State Key Lab on Intelligent Technology and Systems,
National Lab for Information Science and Technology, Tsinghua University, China
chenao3220@gmail.com, sms@mail.tsinghua.edu.cn

Abstract

With the explosive growth of Internet, more and more domain-specific environments appear, such as forums, blogs, MOOCs and etc. Domain-specific words appear in these areas and always play a critical role in the domain-specific NLP tasks. This paper aims at extracting Chinese domain-specific new words automatically. The extraction of domain-specific new words has two parts including both new words in this domain and the especially important words. In this work, we propose a joint statistical model to perform these two works simultaneously. Compared to traditional new words detection models, our model doesn't need handcraft features which are labor intensive. Experimental results demonstrate that our joint model achieves a better performance compared with the state-of-the-art methods.

1 Introduction

Accompanying with the development of Internet, many new specific domains appear, such as forums, blogs, Massive Open Online Courses (MOOCs) and etc. There are always a group of important words in these domains, which are known as domain-specific words. Domain-specific words include two types as shown in Table 1. The first ones are rare and unambiguous words which will seldom appear in other domains such as "栈顶"(stack top) and "二叉树"(binary tree). These words may cause word segmentation problems. For example, if we do not recognize "栈顶"(stack top) as a word, the segmentation "栈 顶 运算符 是 乘号"(the operator at stack top is multiplication sign) will be like "栈 顶运 算符 是 乘号". In this case, "栈顶" means "stack top" and "运算符" means "operator", but in the segmentation result, "顶运" is segmented into a word in mistake and will bring lots of problems to the further applications.

Domain words	Translation	Type
栈顶	stack top	1
二叉树	binary tree	1
复杂度	complexity	2
遍历	iterate	2

Table 1: Examples of domain-specific word in data structure domain

The other type is common and ambiguous words which have specific new meanings in this domain, such as "复杂度"(complexity) and "遍历"(iterate). These words often play important roles in domain-specific tasks. For example, in MOOCs which are typical domain-specific environments, there is an Automated Navigation Suggestion(ANS)(Zhang et al., 2017) task which suggests a time point for users when they want to review the front contents of the video. With the help of the recognition of this type of words, we can easily give higher weights to those domain-specific contents.

After extracting these two type of words, we can also use them for creating ontologies, term lists, and in the Semantic Web Area for finding novel entities(Färber et al., 2016). Besides, in MOOCs area it will also benefit Certification Prediction(CP)(Coleman et al., 2015) (which predicts whether a user will get a course certification or not), Course Recommendation(CR)(Aher and Lobo, 2013) and so on by providing textual knowledge.

Researchers have made great efforts to extract domain-specific words. Traditional new word detection methods usually employ statistical methods according to the pattern that new words ap-

*Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 44–53,
Vancouver, Canada, August 3-4, 2017.

pear constantly. Such methods like Pointwise Mutual Information(Church and Hanks, 1990), Enhanced Mutual Information(Zhang et al., 2009), and Multi-word Expression Distance(Bu et al., 2010). These methods focus on extracting the first type of domain-specific words and conduct post-processing to discover the second type of words. Deng et al. proposed a statistical model TopWords(Deng et al., 2016) to extract the first type of words, it can imply some of these statistical measures into the model itself. Besides, it designs a feature called relative frequency to extract the second type of domain-specific words. TopWords is based on a Word Dictionary Model(WDM)(Ge et al., 1999; Chang and Su, 1997; Cohen et al., 2007) in which a sentence is sampled from a word dictionary. To extract the second type of words, it needs to train its model on a common background corpus which is expensive and time-consuming.

To address these issues, we propose a Domain TopWords model by assuming that a sentence is sampled from two word dictionaries, one for common words and the other for domain-specific words. Besides, we propose a flexible domain score function to take the external information into consideration, such as word frequencies in common background corpus. Therefore, the proposed model can extract these two types of words jointly. The main contributions of this paper are summarized as follows:

- We propose a novel Domain TopWords model that can extract both two types of domain-specific words jointly. Experimental results demonstrate the effectiveness of our model.
- Our model achieves a comparable performance even with much less information comparing to the origin TopWords model.

The rest of the paper is structured as follows: the related work will be introduced in section 2. Our model will be introduced in section 3, including model definition and the algorithm details. Then we will present the experiments in section 4. Finally, the work is summarized in section 5.

2 Related work

New word detection as a superset of new domain-specific word detection has been investigated for a long time. New word detection methods mainly contain two directions: the first ones conduct the word segmentation and new word detection jointly. Most of them are supervised models, typical models include conditional random fields proposed by Peng et al. (2004). These supervised models cannot be used in domain-specific words detection directly, due to the lack of annotated domain-specific data. In addition, there are also some unsupervised models, such as TopWords proposed by Deng et al. (2016). However, it needs time-consuming post-processing to extract the second type of domain-specific words.

Another type treats new word detection as a separate task. This line of methods can be mainly divided into three genres. The first genre is usually preceded by part-of-speech tagging, and treats the new word detection task as a classification problem or directly extracts new words by semantic rules. For example, Argamon et al. (1998) segments the POS sequence of a multi-word into small POS tiles, and then counts tile frequency in both new words and non-new words on training sets, then uses these counts to extract new word. Chen and Ma (2002) uses statistical rules to extract new Chinese word. GuoDong (2005) proposes a discriminative Markov Model to detect new words by chunking one or more separated words. However, these supervised models usually need expert knowledge to design linguistic features and lots of annotated data which are expensive and unavailable in the new arising domains.

The second genre employs user behavior data to detect new words. User typing behavior in Sogou Chinese Pinyin input method which is the most popular Chinese input method is used to detect new words by Zheng et al. (2009). Zhang et al. (2010) proposed to utilize user query log to extract new words. However, these works are usually limited by the availability of the commercial resources.

The third genre employs statistical features and has been extensively studied. In this type of works, new word detection is usually considered as multi-word expression extraction. The measurements of multi-word association are crucial in this type of work. Traditional measurements include: Pointwise Mutual Information (PMI) (Church and Hanks, 1990) and Symmetrical Conditional Probability (SCP) (da Silva and Lopes, 1999). Both these two measures are proposed to measure bi-gram association. Among all 84 bi-

gram association measurements, PMI has been reported to be the best in Czech data(Pecina, 2005). To measure arbitrary of n-grams, some works separate n-grams into two parts and adopt the existing bi-gram based measurements directly. Some other n-gram based measures are also proposed, such as Enhanced Mutual Information (EMI) Zhang et al. (2009). And Multi-word Expression Distance (MED) was proposed by Bu et al. (2010) which based on the information distance theory. The MED measure was reported superior performance to EMI, SCP and other measures. And a pattern based framework which integrates these statistical features together to detect new words was proposed by Huang et al. (2014).

3 Methodology

In this section, we propose a Domain TopWords model. We introduce the Word Dictionary Model(Ge et al., 1999; Chang and Su, 1997; Cohen et al., 2007) and TopWords model proposed by Deng et al. (2016) in subsection 3.1 and 3.2. Then we introduce our Domain TopWords model in subsection 3.3, 3.4 and 3.5. At last, we introduce the modified EM algorithm for our model in 3.6.

3.1 Word Dictionary Model

Word Dictionary Model (WDM) is a unigram language model. It treats a sentence as a sequence of basic units, i.e., words, phrases, idioms, which in this paper are broadly defined as "words". Let $D = \{w_1, w_2, \ldots, w_N\}$ be the vocabulary (dictionary) which contains all interested words, then the sentence can be represented as $S_i = w_{i_i} w_{i_2} \ldots w_{i_j}$. And each word is a sequence of characters. Let $A = \{a_1, \ldots, a_p\}$ be the basic characters of the interested language which in English contain only 26 letters but may include thousands of distinct Chinese characters. Then the words can be represented as $w_i = a_{i_1} a_{i_2} \ldots a_{i_j}$. WDM treats each sentence S as a sampling of words from D with the sampling probability θ_i for word w_i. Let $\theta = (\theta_1, \theta_2, ...\theta_N)$ be the sampling probability of the whole D, then the probability of sampling a specific sentence with length K is:

$$P(S|D,\theta) = \prod_{k=1}^{K} \theta_k \tag{1}$$

3.2 TopWords

TopWords algorithm based on WDM is introduced in Deng et al. (2016), and is used as an unsupervised Chinese text segmentation and new word discovery method. In English texts, words are split by spacing, but in Chinese, there is no spacing between words in a sentence. For unsegemented Chinese text T, let C_T denote the set of all possible segmentations under the dictionary D. Then, under WDM, we have the probability of a Chinese text T:

$$P(T|D,\theta) = \sum_{S_i \in C_T} P(S_i|D,\theta) \tag{2}$$

Then the likelihood of the parameter θ under the given corpus G is:

$$\begin{aligned} L(\theta|D,G) &= P(G|D,\theta) \\ &= \prod_{T_j \in G} P(T_j|D,\theta) \\ &= \prod_{j=1}^{n} \sum_{S_i \in C_{T_j}} P(S_i|D,\theta) \end{aligned} \tag{3}$$

where θ_{i_k} is the sampling probability of k-th word w_{i_k} in segmentation S_i, n is the number of sentences in the corpus G. Then the value of θ can be estimated by the maximum-likelihood estimate(MLE) as follows:

$$\theta^* = \arg\max_{\theta} \prod_{j=1}^{n} \sum_{S_i \in C_{T_j}} P(S_i|D,\theta) \tag{4}$$

The MLE value of θ can be computed by the EM algorithm.

After extracting the first type of domain-specific words, the author proposes a measure called relative frequency to extract the second type of domain-specific words. The relative frequency ϕ_i^k of word w_i in domain k can be estimated as follows:

$$\phi_i^k = \frac{\theta_i^k}{\sum_{j=1}^{K} \theta_i^j} \tag{5}$$

θ_i^k is estimated probability of word w_i from the kth domain.

3.3 Domain Word Dictionary Model

To add the ability to discover domain-specific words, we first use a Domain Word Dictionary Model (D-WDM) instead of the origin WDM model. D-WDM regards a sentence as a sampling from two word dictionaries, one is the common background word dictionary D^c and the other is the domain word dictionary D^d. So a word w_i in a sentence S is sampling first with probability φ to determine which dictionary it is from, and then with probability θ_i^ι from D^d or D^c. So the probability of sampling in D-WDM a specific sentence with length K is:

$$P(S_i|D,\theta,\varphi) = \prod_{k=1}^{K_i}(\varphi\theta_{i_k}^c + (1-\varphi)\theta_{i_k}^d) \quad (6)$$

where

$$\theta = (\theta^c, \theta^d) \quad (7)$$

3.4 Domain TopWords

The main difference between Domain TopWords(D-TopWords) and TopWords is that D-TopWords is under the D-WDM model. So there are two word dictionaries, one for common words and the other for the domain-specific words. So the likelihood of θ with the given corpus G under the D-WDM model is:

$$\begin{aligned} L(\theta|D,G,\varphi) &= \prod_{T_j\in G}\sum_{S_i\in C_{T_j}} P(S_i|D,\theta,\varphi) \\ &= \prod_{j=1}^{n}\sum_{S_i\in C_{T_j}}\prod_{k=1}^{K_i}(\varphi\theta_{i_k}^c + (1-\varphi)\theta_{i_k}^d) \end{aligned} \quad (8)$$

where the parameter φ need to be fixed. If the φ is adapted, the model will converge at a point which maximize the probability difference of the words between the initial θ_d and θ_c.

However, in the D-WDM model, there is no difference between the domain dictionary D_d and the common dictionary D_c except the parameter φ. So if we use pure EM algorithm to estimate the parameter θ^c and θ^d, it is obvious that the algorithm cannot determine whether a word should be sampled from D_c or D_d. And even though the model has the ability to distinguish the two kinds of words, it can not find out which words are domain-specific words either if we only use the domain-specific corpus. So we must add the common background corpus knowledge into our model and denote this function as domain score function σ.

Domain TopWords model uses an optimized probability function of a segmentation which can take the background knowledge into consideration. The probability of a segmentation S_i of a sentence as follows:

$$P(S_i|T;D,\theta,\varphi,\sigma) = \frac{Q(S_i|T;D,\theta,\varphi)}{\sum_{S_i\in C_T} Q(S_i|T;D,\theta,\varphi)} \quad (9)$$

$$Q(S_i|T;D,\theta,\varphi,\sigma) = \prod_{k=1}^{K_i}(\varphi\theta_{i_k}^c + (1-\varphi)\theta_{i_k}^d\sigma_{i_k}) \quad (10)$$

is the score of the sampled segmentation S_i of T. $P(S_i|T;D,\theta,\varphi,)$ is the nomorlized version of $Q(S_i|T;D,\theta,\varphi,\sigma)$. σ_{i_k} is the domain score of the word w_{i_k}.

3.5 Selection of domain score σ

As mentioned above, we need a domain score function σ to tell our model how to distinguish whether a word is a common word or a domain-specific word. This function has several choices, i.e., the frequency of the word in a large background corpus, matches of specific templates, and so on. And we find that statistical features, like left(right) entropy and mutual information, are useless as the background knowledge function because the D-TopWords model itself has taken this part of features into consideration. We introduce some choices of the σ function and evaluate the effects in our experiment.

Constant Score The first choice of σ function is a constant function which returns a constant number for all words. This means there is no encouragement for any word so that we will get a θ^d which has almost the same word distribution as θ^c. We denote D-TopWords with constant σ function as D-TopWords+Const.

Background Frequency Score It is a natural idea that uses the reciprocal of the frequency of the word in a common background corpus. This σ function encourages words with low background frequency to be sampled from θ^d. The detailed

function is as follows:

$$\sigma(w) = \sqrt{\frac{P}{Fre(w)}} \tag{11}$$

where P is a constant. The parameter P need to be tuned according to the size of the domain corpus, in our experiments we choose 900 to get a domain score in the range of 1-10 for domain words. And $Fre(w)$ is the frequency of word w in background corpus. We denote the result as D-TopWords+Fre.

RF Score We use the reciprocal of word probability in the dictionary of the origin TopWords method estimated with common background corpus as our domain score. We denote this function as RF function respect to the relative frequency in TopWords. The detailed function is as follows:

$$\sigma(w) = \sqrt{\frac{1}{WP(w) \times 10^5}} \tag{12}$$

where the $WP(w)$ is the word probability of word w in the dictionary of origin TopWords model. We denote the result as D-TopWords+RF.

3.6 EM estimation of θ

The parameter θ will be estimated by the EM algorithm as we will show below. In the beginning, we add all the words in vocabulary to θ and default values will be set for both θ^c and θ^d before EM steps. We employ a "top-down" strategy to discover words, and this is the reason why this method is called TopWords. It adds all words into its dictionary at first and then drops the words whose probability is close to zero (e.g., $< 10^{-8}$, and we use this value in our experiments). A good choice of the default value for θs is the normalized frequency vector of the words in the corpus.

Next, we will show the EM algorithm for our D-TopWords model. Let $\theta^{(r)}$ be the estimated value of θ at the r-th iteration. Then the E-step and the M-step can be computed as follows. The E-step computes the Q-function:

$$\begin{aligned} Q(\theta|\theta^{(r)}) =& E_{S|G,\theta^{(r)}}[logL(\theta; G, S)] \\ =& \sum_{j=1}^{n} \sum_{S \in C_{T_j}} P(S|T_j; D, \theta^{(r)}) \\ & logP(S|D, \theta) \end{aligned} \tag{13}$$

and the M-step maximizes $Q(\theta|\theta(r))$ so as to update θ_d and θ^c as follows

$$\begin{aligned} \theta^{c(r+1)} &= (c_1^{(r)}, \ldots, c_N^{(r)}, n)/(n + \sum_i (c_i^{(r)})) \\ \theta^{d(r+1)} &= (d_1^{(r)}, \ldots, d_N^{(r)}, n)/(n + \sum_i (d_i^{(r)})) \end{aligned} \tag{14}$$

where

$$\begin{aligned} c_i^{(r)} &= \sum_{T_j \in G} c_i(T_j) \\ c_i(T_j) &= \sum_{S \in T_j} c_i(S) \cdot P(S|T_j; D, \theta^{(r)}) \cdot \\ & \qquad \frac{\varphi \theta_i^{c(r)}}{\varphi \theta_i^{c(r)} + (1-\varphi)\theta_i^{d(r)} \sigma_i} \end{aligned} \tag{15}$$

$c_i(S)$ is the number of occurrences of w_i which is sampled from common dictionary in sentence S, and

$$\begin{aligned} d_i^{(r)} &= \sum_{T_j \in G} d_i(T_j) \\ d_i(T_j) &= \sum_{S \in T_j} d_i(S) \cdot P(S|T_j; D, \theta^{(r)}) \cdot \\ & \qquad \frac{(1-\varphi)\theta_i^{d(r)}}{\varphi \theta_i^{c(r)} + (1-\varphi)\theta_i^{d(r)} \sigma_i} \end{aligned} \tag{16}$$

$d_i(S)$ is the number of occurrences of w_i which is sampled from domain dictionary D_d in sentence S.

In the experiment, we found that because of the lack of domain-specific data the model tends to get long words and short segmentation. We add a segmentation length related factor to reduce this tendency, then our Q function of segmentation S_i becomes:

$$Q(S_i|\theta) = \alpha^{K_i} \prod_{k=1}^{K_i} (\varphi \theta_{i_k}^c + (1-\varphi)\theta_{i_k}^d \sigma_{i_k}) \tag{17}$$

α is a constant parameter. K_i is the length of the segmentation S_i.

4 Experiments

In this section, we first perform an experiment to compare our method to several baselines. And

$top\ K\ words \Rightarrow$	**100**	**200**	**400**	**700**
Huang et al.(2014)	0.435	0.413	0.378	0.353
D-TopWords+Const	0.266	0.162	0.152	0.150
TopWords+Fre	0.630	0.576	0.495	0.412
D-TopWords+Fre	0.719	0.664	0.573	0.504
TopWords+RF	0.759	0.679	0.601	0.548
D-TopWords+RF	**0.795**	**0.705**	**0.615**	**0.553**

Table 2: Discovering new words in data structure domain (MAP)

then we perform parameter analysis to demonstrate how the parameters will affect our model. At last, we conduct some case studies to analysis these methods in details.

4.1 Data Preparation

We use transcripts of an online course called Data Structure from Xuetangx.com. Xuetangx.com is one of the biggest MOOC platforms in China. These transcripts are a total of 55,045 lines, including 655312 Chinese characters in it and totally 1,792 different characters.

We segment the corpus by characters and count the frequency of character-based n-grams from unigram up to 7-gram. We drop words with the frequency less than 5 and result in a 55,452 lines n-gram list. The resulted n-gram list is very sparse (close to 1:170) and most of the results are obviously meaningless (like "这样一" which means "one such"). We asked two annotators to label these n-grams. These two annotators are requested to judge whether an n-gram is a domain-specific word or not, it takes almost one week to annotate these n-grams. If there is a disagreement in these annotations, the annotators will discuss the final annotation and result in a 12.6% disagreement ratio. Most of the disagreements are like "访问"(visit) and "插入"(insert) which are somewhat ambiguous. Finally, we use a relatively strict standard, this results in 326 domain-specific words. The final annotated file can be accessed in our Github repo[1].

We use YUWEI corpus as our common background corpus. This corpus is developed by the National Language Commission, which contains 25,000,309 words with 51,311,659 characters.

4.2 Evaluation Metric

The output of our method is a ranked list, so we use mean average precision (MAP) as one of our evaluation metrics. The MAP value is computed as follows:

$$MAP(K) = \frac{\sum_{k=1}^{K} P(k) \times rel(k)}{\sum_{k=1}^{K} rel(k)} \quad (18)$$

where the $P(k)$ is the precision of the top k words, $rel(k)$ is a indicator function which return 1 when word at rank k is a domain-specific word and 0 otherwise. K is the length of the result list. When we get a list whose elements are all domain-specific words, the $MAP(K)$ will be 1.

We will also display the precision-recall curves of our results.

4.3 Discovering New Words

4.3.1 Experiment Settings

We compare different settings of our method with two baselines. The first baseline is pattern-based unsupervised new word detection method, which is proposed by Huang et al. (2014). The following statistical features are taken into consideration: left pattern entropy (LPE), normalized multi-word expression distance (NMED), enhanced mutual information (EMI). We implement both character based and word-based version, and the word-based version outperforms character based version. We use the optimal parameter setting in Huang's method, which is the LPE+NMED setting in their paper. And we use annotated words to extract the candidate patterns which is a pretty good treatment for this method.

The second baseline is origin TopWords method which has been mentioned in above section. We first run the TopWords method in the domain-specific corpus, and then use a function to rerank the word dictionary θ. We use two functions to rerank the dictionary. The first one is the background frequency function and we denote this version as TopWords+Fre. The second one is the standard relative frequency method, we use the dictionary θ_B of TopWords method run in background

[1] http://github.com/dreamszl/dtopwords

D-TopWords+Fre	TopWords+Fre	Huang et al.
具体来说(specifically speaking)	接下来(next)	确实(indeed)
请注意(attention please)	换而言之(in other words)	至少(at least)
换而言之(in other words)	具体来说(specifically speaking)	对齐位置(alignment position)
字符(character)	同学们好(hello students)	顺序性(succession)
括号(brackets)	我们(we)	诸如此类(and so on)

Table 3: Top 5 wrong results of D-TopWords+Fre, TopWords+Fre and Huang et al.'s method

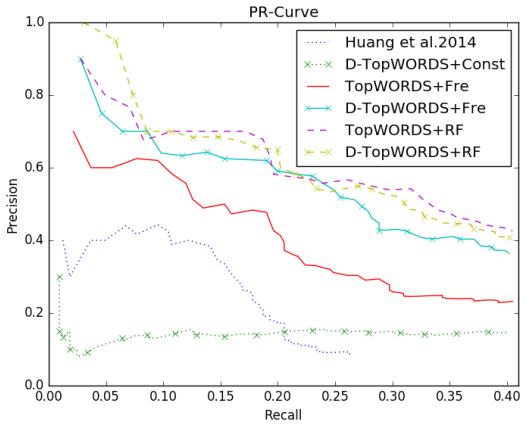

Figure 1: PR-Curves of our methods and two baselines

corpus to rerank θ. We denote this version as TopWords+RF.

4.3.2 Result and Analysis

(1) The MAP values of all the methods are shown in Table 2, and the PR-curves are shown in Figure 1. From the results, we can see our D-TopWords+RF and TopWords+RF achieve the best performance. Our D-TopWords+RF achieves better performance than TopWords+RF method, especially when the recall is lower our D-TopWords+RF outperforms TopWords+RF obviously as shown in Figure 1. In the actual application scenario, our model is more practical as the top results returned by the model are more important.

(2) Our D-TopWords methods achieve better performance than the corresponding TopWords results. We expect that our D-TopWords model can use the external information more effectively and accurately. Our D-TopWords model will give more weights to the probability whether a sequence can be a word or not, and the TopWords model will more reliable on the external information.

(3) More than that, our D-TopWords+Fre methods is significantly better than TopWords+Fre model and comparable to the D-TopWords+RF and TopWords+Rf model. The external background information RF takes the probability a sequence can be a word or not into consideration, however, our D-TopWords can consider this information in the model itself. So RF information is relative redundancy than Fre information to our D-TopWords model. The RF information needs to be trained on the common background corpus when the common background corpus is large it will take a very long time.

(4) We perform experiments of Huang et al.'s method with different domain score functions and all of these result in a poor performance. With the recall raising the precision decreases sharply, we suppose that it is because such statistical features based models cannot deal with low-frequency words well. However, our model can deal with this kind of words better by using the context information. And our model can hold a better balance between the probability whether a sequence can be a word or not and the domain score, which is hard for Huang et al.'s method.

4.4 Parameter Tuning

Table 5 shows how the performance changes with different α which is the segmentation length related parameter and φ which is the dictionary weight parameter. As we can see, the performance gets better when φ increases and get the best result when φ is 0.9. φ represents the probability a word is sampled from the common dictionary, so it means that a word is sampled from the common dictionary with a 90% possibility and domain-specific dictionary with 10%.

It achieves the best performance when φ is set as 0.9 and α is set as 100. Looking into the results, we found α determines the length of the words in θ. When α chooses a smaller value the results tend to be longer, when α chooses bigger value the results tend to be shorter. And when the size of cor-

Data Structure	*University Chemistry*	*Nuclear Physics*
遍历(iterate)	过程当中(in the process)	衰变(decay)
关键码(key code)	平衡常数(equilibrium constant)	活度(activity)
递归(recursive)	配合物(complex)	放射源(redioactive source)
具体来说(specifically speaking)	解离(dissociation)	γ 射线(γ-ray)
复杂度(complexity)	吉布斯自由能(Gibbs free energy)	去表示(to express)
BST (binary search tree)	杂化轨道(hybrid track)	入射粒(incident grain)
左孩子(left child)	孤对电子(lone paired electron)	MeV (MeV)
运算符(operator)	电极电势(electrode potential)	靶核(target nucleus)
数据结构(data structure)	同学们好(hello students)	半衰期(half-life period)
B 树(B tree)	反应速率(reaction rate)	核素(species)

Table 4: Top 10 results of D-TopWords+Fre in three courses

φ \ α	10	50	100	500	1000
0.3	0.243	0.344	0.389	0.416	0.429
0.5	0.323	0.441	0.479	0.529	0.516
0.7	0.405	0.513	0.559	0.593	0.483
0.9	0.437	0.672	0.719	0.547	0.448
0.99	0.306	0.470	0.479	0.519	0.447

Table 5: MAP of top 100 results'performance with different α and φ, under the D-TopWords+Fre model.

pus increasing, a smaller α value will get better performance. We set α as 10 when estimates θ of the common background corpus.

4.5 Case study

(1) The top five wrong results of D-TopWords+RF and TopWords+RF are similar. There are some wrong results appearing in top 100 results in TopWords+RF but not in D-TopWords+RF such as "大家注意"(everybody attention). After inspecting the common dictionary θ_c in D-TopWords+RF, we find both "大家"(everybody) and "注意"(attention) are in high ranks. We suppose that the usage of Domain Word Dictionary Model helps to deal with this type of sequences better.

(2) The teacher of this course uses "换而言之"(in other words), "具体来说"(specifically speaking) very frequently, so the TopWords+Fre and D-TopWords+Fre cannot recognize them. And the wrong results "接下来"(next) and "同学们好"(hello students) rank lower in our method compared to TopWords+Fre method (i.e., 25 and 41 vs 4 and 13). We suppose that it is because our method can keep a better balance of the domain score and the probability that a sequence be a new word. And we inspect other wrong results which have a similar situation, these words all have a much lower rank in our method. So these phenomena confirm our assumption that our model achieves better performance in the sequences that with low frequency in background corpus but cannot be a word.

(3) The wrong result "我们"(we) doesn't appear in the domain dictionary θ_d, but appears at rank 7 in the θ_c dictionary in our model. There are also some results appearing in a high rank in TopWords+Fre method, but in a low rank in our D-TopWords+Fre method. For example, 比如说(for example) ranks in 39 in TopWords+Fre but rank in 574 in D-TopWords+Fre, "这么样"(the same as it) ranks in 31 in TopWords+Fre but ranks in 2759 in D-TopWords+Fre, "也就是"(that's it) ranks in 53 in TopWords+Fre but not appear in our method, and so on. We suppose that the usage of Domain Word Dictionary Model is the reason that our model can reach a better performance in these type of words.

(4) The first 10 results (D-TopWords+Fre) in *Data Structure* course and two other courses are shown in table 4.

5 Conclusion

We propose a pure unsupervised D-TopWords model to extract new domain-specific words. Compared to traditional new word extraction model, our model doesn't need handcrafted lexical features or statistical features and starts from the unsegmented corpus. Compared to the origin TopWords model, our model can reach a better performance with the same information and can reach a comparable performance with only back-

ground corpus frequency information to the TopWords model with the relative frequency which is expensive and time-consuming.

Our D-TopWords model adds the ability to distinguish whether a word from common dictionary or domain dictionary to the origin TopWords model. We add a domain score parameter to let our model which can take the external information easily and efficiently. Experiments show that due to our modification our model can use much less external information to reach a comparable performance to the origin TopWords model.

Ackonwledgements

I am very grateful to my friends in THUNLP lab and the reviewers for giving many suggestions in the course of my thesis writing. This work is supported by Center for Massive Online Education, Tsinghua University, and XuetangX (http://www.xuetangx.com/), the largest MOOC platform in China.

References

Sunita B Aher and LMRJ Lobo. 2013. Combination of machine learning algorithms for recommendation of courses in e-learning system based on historical data. *Knowledge-Based Systems* 51:1–14.

Shlomo Argamon, Ido Dagan, and Yuval Krymolowski. 1998. A memory-based approach to learning shallow natural language patterns. In *Proceedings of the 17th international conference on Computational linguistics-Volume 1*. Association for Computational Linguistics, pages 67–73.

Fan Bu, Xiaoyan Zhu, and Ming Li. 2010. Measuring the non-compositionality of multiword expressions. In *Proceedings of the 23rd International Conference on Computational Linguistics*. Association for Computational Linguistics, pages 116–124.

Jing-Shin Chang and Keh-Yih Su. 1997. An unsupervised iterative method for chinese new lexicon extraction. *Computational Linguistics and Chinese Language Processing* 2(2):97–148.

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*. Association for Computational Linguistics, pages 1–7.

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

Paul Cohen, Niall Adams, and Brent Heeringa. 2007. Voting experts: An unsupervised algorithm for segmenting sequences. *Intelligent Data Analysis* 11(6):607–625.

Cody A Coleman, Daniel T Seaton, and Isaac Chuang. 2015. Probabilistic use cases: Discovering behavioral patterns for predicting certification. In *Proceedings of the Second (2015) ACM Conference on Learning@ Scale*. ACM, pages 141–148.

J Ferreira da Silva and G Pereira Lopes. 1999. A local maxima method and a fair dispersion normalization for extracting multi-word units from corpora. In *Sixth Meeting on Mathematics of Language*. pages 369–381.

Ke Deng, Peter K Bol, Kate J Li, and Jun S Liu. 2016. On the unsupervised analysis of domain-specific chinese texts. *Proceedings of the National Academy of Sciences* page 201516510.

Michael Färber, Achim Rettinger, and Boulos El Asmar. 2016. On emerging entity detection. In *Knowledge Engineering and Knowledge Management: 20th International Conference, EKAW 2016, Bologna, Italy, November 19-23, 2016, Proceedings 20*. Springer, pages 223–238.

Xianping Ge, Wanda Pratt, and Padhraic Smyth. 1999. Discovering chinese words from unsegmented text (poster abstract). In *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. ACM, pages 271–272.

Zhou GuoDong. 2005. A chunking strategy towards unknown word detection in chinese word segmentation. In *International Conference on Natural Language Processing*. Springer, pages 530–541.

Minlie Huang, Borui Ye, Yichen Wang, Haiqiang Chen, Junjun Cheng, and Xiaoyan Zhu. 2014. New word detection for sentiment analysis. In *ACL (1)*. pages 531–541.

Pavel Pecina. 2005. An extensive empirical study of collocation extraction methods. In *Proceedings of the ACL Student Research Workshop*. Association for Computational Linguistics, pages 13–18.

Fuchun Peng, Fangfang Feng, and Andrew McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *Proceedings of the 20th international conference on Computational Linguistics*. Association for Computational Linguistics, page 562.

Han Zhang, Maosong Sun, Xiaochen Wang, Zhengyang Song, Jie Tang, and Jimeng Sun. 2017. Smart jump: Automated navigation suggestion for videos in moocs. In *Proceedings of the 26th International Conference on World Wide Web Companion*. International World Wide Web Conferences Steering Committee, pages 331–339.

Wen Zhang, Taketoshi Yoshida, Xijin Tang, and Tu-Bao Ho. 2009. Improving effectiveness of mutual information for substantival multiword expression extraction. *Expert Systems with Applications* 36(8):10919–10930.

Yan Zhang, Maosong Sun, and Yang Zhang. 2010. Chinese new word detection from query logs. In *International Conference on Advanced Data Mining and Applications*. Springer, pages 233–243.

Yabin Zheng, Zhiyuan Liu, Maosong Sun, Liyun Ru, and Yang Zhang. 2009. Incorporating user behaviors in new word detection. In *IJCAI*. Citeseer, volume 9, pages 2101–2106.