Journal of Computational Information Systems 9: 16 (2013) 6603–6611
Available at http://www.Jofcis.com

Similarity Measure for Short Texts Using Topic Models and Rough Sets

Zhifei ZHANG [1,*], Duoqian MIAO [1], Xiaodong YUE [2]

[1] *Department of Computer Science and Technology, Tongji University, Shanghai 201804, China*

[2] *School of Computer Engineering and Science, Shanghai University, Shanghai 200444, China*

Abstract

Synonymy and polysemy are two key problems in short text analysis. Most existing methods based on topic models have limitations. In this paper, we propose a two-stage method based on topic models and rough sets. The topics are firstly generated using latent Dirichlet allocation. Second, we adjust the weights of synonymous and polysemous words discovered by rough sets. The experimental results show the proposed method can achieve good performances on both English and Chinese short texts.

Keywords: Short Texts; Similarity Measure; Topic Models; Rough Sets

1 Introduction

Texts can be classified into long texts and short ones according to length. As a carrier, social media allows users to post short texts quickly and easily. Each tweet on Twitter or each microblog post is limited to 140 characters. The length of user status on Facebook and Renren is restricted to 420 and 240 characters respectively. Short texts are common in search engine [1], question answering [2] and so on.

Synonymy and polysemy are highlighted in short texts [3]. The former means that distinct words are of the same meaning, and the latter means that the same word is of different meanings. The short length, pervasive abbreviations, and buzzwords exacerbate the two issues [4]. The performances of traditional text representation, e.g., vector space model (VSM for short) [5], and machine learning methods directly used in short texts are not effective [6, 7].

Many studies have been done to enrich text representation with external resources. The resources include search results from search engines [8, 9], and knowledge bases (e.g., WordNet and Wikipedia) [10, 11]. Topic models have been introduced to measure similarity more accurately. If two short texts are both related to one certain topic, they are considered to be similar. Latent Dirichlet allocation (LDA for short) [12] is a typical topic model which can generate latent topics. Phan et al. [6] represent texts with topics extracted from Wikipedia. Quan et al. [13] discover the internal relation of words based on topics.

*Corresponding author.
Email address: zhifei.zzh@gmail.com (Zhifei ZHANG).

1553–9105 / Copyright © 2013 Binary Information Press
DOI: 10.12733/jcisP0764
August 15, 2013

Table 1: Comparison of existing methods

	Method	Synonymy	Polysemy	Limitation
Search results	[8, 9]	Yes	No	They depend strongly on search results and cost much time.
Knowledge bases	[10, 11]	Yes	No	They will become useless if words do not appear in these bases.
Topic models	[6]	Yes	Yes	It has to run LDA again for testing and can not measure similarity directly.
	[13]	Yes	No	It needs a predefined threshold to find synonym.

The above methods have their limitations. We focus on topic models. Although the method in [6] can solve the problem of synonymy and polysemy, it is not suitable for measuring the similarity of two short texts directly. The method in [13] can compare a pair of short texts directly, however, it only solves the problem of synonymy with a predefined threshold. Thus, we propose a new approach which can not only measure similarity directly for two short texts, but also solve the problem of synonymy and polysemy without extra parameters.

The rest of this paper is organized as follows. We briefly review the background in the next section. Section 3 presents the details of our method. Experimental designs and results are shown in Section 4. Finally, we draw a conclusion in Section 5.

2 Background

2.1 Topic models

Blei et al. proposed latent Dirichlet allocation [12] in 2003. Each text is a random mixture of latent topics, and each topic is a probabilistic distribution of words. Griffiths et al. [15] introduced another hyper-parameter to make the topic-word distribution obey Dirichlet distribution.

The graph model of LDA is illustrated in Fig. 1. In this figure, θ is the text-topic distribution, ϕ is the topic-word distribution, α is the hyper-parameter for θ, β is the hyper-parameter for ϕ, z is the topic assignment for a word, M is the number of texts, N is the number of words, and T is the number of topics.

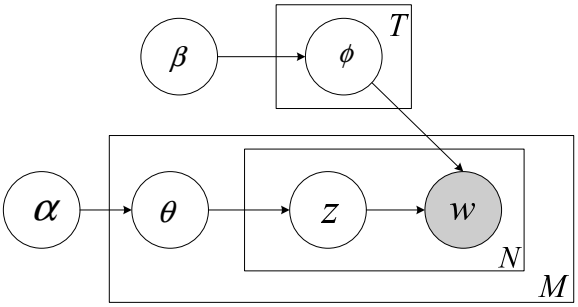

Fig. 1: Graph model of LDA

The hyper-parameters α and β are usually fixed as $\alpha = \frac{50}{T}$ and $\beta = 0.01$ [12]. The parameter ϕ can be estimated via Gibbs sampling [16].

$$\phi_{jk} = \frac{n_j^{(k)} + \beta}{\sum_{i=1}^{N} n_j^{(i)} + N\beta}, \tag{1}$$

where $n_j^{(i)}$ is the total times word v_i is assigned to the j-th topic.

2.2 Rough sets

Pawlak proposed Rough set theory [17] in 1982. It is a useful mathematical tool to deal with imprecise and uncertain data. Rough set theory represents knowledge with an information system and approximates traditional sets with a pair of sets, i.e., lower approximation and upper approximation.

Definition 1 *(Information System) [17] An information system is formalized as a quadruple, $IS=(U, A, VA, f)$, where U is a finite set of objects, A is a finite set of attributes, VA is a set of attribute values, and f is an information function that assigns values from attributes to objects.*

Definition 2 *(Indiscernibility Relation) [17] Given a set $R \subseteq A$, there is an associated equivalence relation, denoted by $IND(R)$ or R. The relation is called an indiscernibility relation.*

$$IND(R) = \{(x, y) \in U^2 | \forall a \in R, a(x) = a(y)\}. \tag{2}$$

Definition 3 *(Lower and Upper Approximations) [17] Given an system $IS=(U, A, VA, f)$, and an indiscernibility relation R, the lower and upper approximations of a set $X \subseteq U$ are defined as:*

$$\underline{R}(X) = \{x \in U | [x]_R \subseteq X\}, \tag{3}$$

$$\bar{R}(X) = \{x \in U | [x]_R \cap X \neq \emptyset\}, \tag{4}$$

where $[x]_R$ is the equivalence class containing x.

The positive region $POS_R(X)$, $POS_R(X)=\underline{R}(X)$, is the complete set of objects that can be certainly classified as belonging to X.

3 The New Method

3.1 Overview of our method

Given two short texts d_1 and d_2 with only three words $\{v_1, v_2, v_3\}$, v_1 and v_3 are synonymous, v_2 is polysemous. Figure 2 describes an overview of our method.

We increase the weights of v_3 in d_1 and v_1 in d_2. It is easy to prove that $\gamma_2 < \gamma_1$, i.e., the similarity of d_1 and d_2 increases because of synonymy. Further, we decrease the weights of v_2 in d_1 and d_2. It is hold that $\gamma_3 > \gamma_2$, i.e., the similarity of d_1 and d_2 decreases because of polysemy.

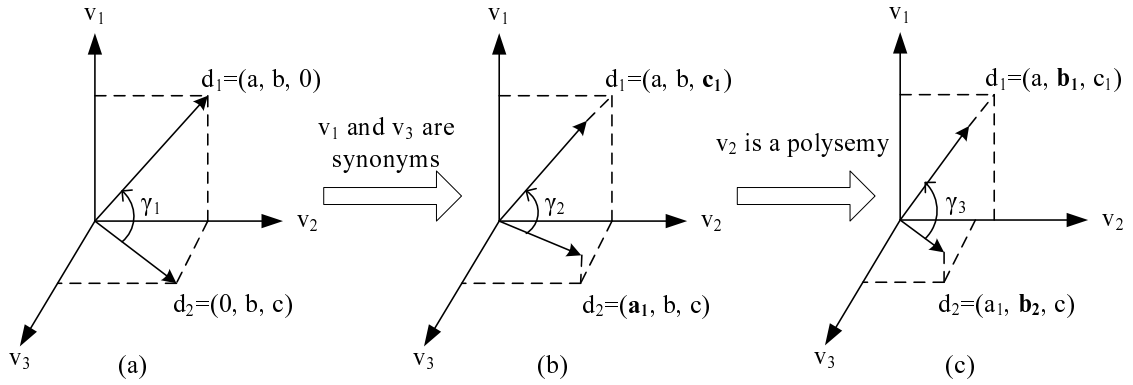

Fig. 2: An overview of our method. γ_1, γ_2 and γ_3 are all vectorial angles. In (b), a_1>0 and c_1>0; In (c), $0<b_1<b$ and $0<b_2<b$.

The method in [13] provides the weight adjusting method from (a) to (b) in Fig. 2 and only solves the problem of synonymy. Our method provides the weight adjusting method from (b) to (c) in Fig. 2 and solves the problem of polysemy.

3.2 Notations

- V: Vocabulary is $V=\{v_1, v_2, \cdots, v_N\}$, where N is the number of unique words.
- D: Text collection is $D=\{d_1, d_2, \cdots, d_M\}$, where M is the number of texts.
- $\mathbf{R}^{(i)}$: Text d_i is represented by a vector $\mathbf{R}^{(i)}=(w_1^{(i)}, w_2^{(i)}, \cdots, w_N^{(i)})$, where $w_k^{(i)}$ is the weight of word v_k in text d_i. For example, two short texts d_1 and d_2 are given:

$$\mathbf{R}^{(1)} = (w_1^{(1)}, w_2^{(1)}, \cdots, w_N^{(1)}), \ \ \mathbf{R}^{(2)} = (w_1^{(2)}, w_2^{(2)}, \cdots, w_N^{(2)}).$$

- Φ: Topic-word distribution estimated via LDA from text collection is represented by a matrix Φ, where ϕ_{jk} indicates the probability of word v_k belonging to the j-th topic.

3.3 Finding synonym and polysemy

The critical step of our method is to find synonym and polysemy for two short texts.

Definition 4 *(Distinguishing Word Sets) [13] Given two short texts d_1 and d_2, the distinguishing word sets of them are defined as:*

$$\begin{aligned} Dist(d_1, d_2) &= \{v | v \in d_1 \wedge v \notin d_2\}, \\ Dist(d_2, d_1) &= \{v | v \in d_2 \wedge v \notin d_1\}. \end{aligned} \tag{5}$$

Definition 5 *(Common Word Set) Given two short texts d_1 and d_2, the common word set of them is defined as:*

$$Comm(d_1, d_2) = \{v | v \in d_1 \wedge v \in d_2\}. \tag{6}$$

Definition 6 *(Topic-Word Information System) Topic-word distribution can be transformed to an information system, $TWIS$=(WS, TS, VA, f), where WS is a set of N words, TS is a set of T topics, VA is a set of topic values, and f is an information function that assigns values from topics to words. For the k-th topic, $f_k : TS \to VA_k$, and VA_k is the range of its value.*

The element of Φ is a real number between 0 and 1. The topic value in $TWIS$ is usually discrete. Therefore, a real number is mapped to a discrete value, e.g., "low", "middle" and "high".

If two words have the similar topic-word distribution, they are considered to be similar under one certain topic. The words in the positive region consistently reflect the related topic. If the words in two positive regions are discernible, the common words may express different topics. For convenience, let $Dist(d_1, d_2)$ be RP, $Dist(d_2, d_1)$ be RQ, and $Comm(d_1, d_2)$ be RC. The procedure of finding synonym and polysemy is described below:

(1) Compute the union set of the above three sets, i.e., RA=$RP \cup RQ \cup RC$;

(2) Extract the subset of $TWIS$ by replacing WS with RA;

(3) Compute two positive regions, $POS_{TS}(RP)$ and $POS_{TS}RQ$;

(4) Find the synonym:

   a. obtain $\Delta RP = RP - POS_{TS}(RP)$ and $\Delta RQ = RQ - POS_{TS}(RQ)$;

   b. for each $p \in \Delta RP$ and $q \in \Delta RQ$, if $(p, q) \in IND(TS)$, then p and q are synonymous;

(5) Find the polysemy:

   a. obtain pairs $\{(p, q)|p \in POS_{TS}(RP)\ and\ q \in POS_{TS}(RQ)\}$;

   b. if exists a pair (p, q) which is not in $IND(TS)$, then $c \in RC$ is polysemous.

3.4 Updating weight

The next step after finding synonym and polysemy is to update the weights of these words.

Assuming that $v_m \in Dist(d_1, d_2)$ and $v_n \in Dist(d_2, d_1)$ are synonymous (m and n are the indexes in V), each of them has the highest probability under the i-th topic, whose value is ϕ_{im} and ϕ_{in} respectively. Their weights in $\mathbf{R}^{(1)}$ and $\mathbf{R}^{(2)}$ are modified as follows [13]:

$$\begin{aligned} w_n^{(1)} &= w_n^{(1)} + w_n^{(2)} \times \phi_{in}, \\ w_m^{(2)} &= w_m^{(2)} + w_m^{(1)} \times \phi_{im}. \end{aligned} \tag{7}$$

Assuming that $v_c \in Comm(d_1, d_2)$ (c is the index in V) is polysemous, the most likely topics for v_c in d_1 and d_2 are obtained as follows:

$$\begin{aligned} u &= \arg\max_s\{\phi_{sk}|1 \le s \le T\ and\ v_k \in POS_{TS}(RP)\}, \\ l &= \arg\max_s\{\phi_{sk}|1 \le s \le T\ and\ v_k \in POS_{TS}(RQ)\}. \end{aligned} \tag{8}$$

Similar to Eq. 7, we adjust the weights of v_c in d_1 and d_2:

$$\begin{aligned} w_c^{(1)} &= w_c^{(1)} - w_c^{(1)} \times \phi_{lc}, \\ w_c^{(2)} &= w_c^{(2)} - w_c^{(2)} \times \phi_{uc}. \end{aligned} \tag{9}$$

4 Experiments

4.1 Data sets

We carry out experiments on two short text data sets: Search-Snippet data set and Web-Title data set. The former is in English and the latter is in Chinese. Their statistics are listed in Table 2.

- **Search-Snippet data set** It is collected by Phan [7], which consists of two subsets, named *Short text* and *Corpus*. *Short text* contains search-snippets selected from the results of web search using predefined phrases of different domains. *Corpus* is formed by the corresponding Wikipedia pages. We want to classify *Short text* using the topics extracted from *Corpus*.
- **Web-Title data set** We collect it by crawling Netease web pages. All titles of pages form the short texts, and the body contents form the background data, named *Title* and *Content* respectively. Each page is belonging to one domain. We want to classify *Title* using the topics extracted from *Content*.

Table 2: Statistics of two data sets

Search-Sinppet	Domain	#Texts	Web-Title	Domain	#Texts
Short text	Business	1500	*Title*	Education	518
	Computer	1500		Economics	701
	Culture-Arts-Ent	2210		Military	1871
	Education-Science	2660		Science and Tech	505
	Engineering	370		Business	501
	Health	1180		Society	483
	Politics-Society	1500		Sports	808
	Sports	1420		Entertainment	505
	Total	12340		**Total**	5892
Corpus	#Words	#Texts	*Content*	#Words	#Texts
	60649	71986		31204	5892

4.2 Experiment settings

- **Generate Topics** *Corpus* and *Content* can be used to generate the suitable number of topics. We determine the number of topics according to perplexity (the smaller, the bet-

ter) [12]. Here, we set the number of topics to 70 on Search-Snippet data set and 50 on Web-Title data set.

- **Performance Evaluation** In terms of performances of KNN and SVM, we compare our method with VSM and the method in [13]. The number of neighbors in KNN is set to be the number with which VSM achieves the best performance. Here, we fix the number of neighbors on two data sets as 21 and 17. SVM is implemented based on libSVM [18] with default setting. Besides, F_1 [16] is used for evaluating classification performance.

4.3 Experiment results

4.3.1 Classification performance

The classification results are described in Fig. 3. The error bars represent one standard deviation determining whether differences are statistically significant. Figure 3 obviously tells us our method outperforms the other two. The performance of our method outperforms the method in [13] by about 3.7% on Search-Snippet data set, and 3.5% on Web-Title data set.

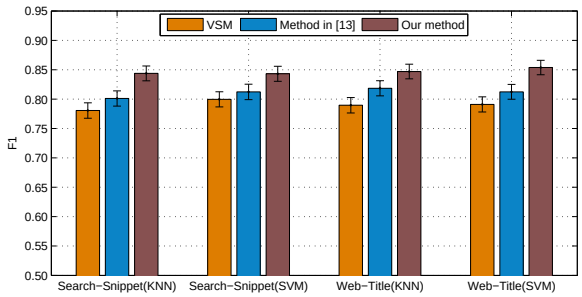

Fig. 3: Classification performance with three different similarity measures

4.3.2 Experimental analysis

In order to explain why the performance is improved, the synonymous and polysemous occurrences are listed in Table 3. The table tells that there really exists a certain quantity of polysemy in short texts. The weights of polysemous words are decreased, which will bring about more accurate similarity measure for short texts.

Table 3: Statistics of synonymous and polysemous occurrences

Data set	Synonymous occurrences	Polysemous occurrences
Search-Snippet	1225	731
Web-Title	983	552

5 Conclusions

This paper proposes a new similarity measure for short texts using topic models and rough sets. We find synonym and polysemy via rough sets on the topic-word distribution generated by latent Dirichlet allocation, and adjust the weights of these words creatively. The experiment results show that our method is better than the other two baselines. How to further improve the performance is left to explore in the future.

Acknowledgement

This work is partially supported by the National Nature Science Foundation of China (Granted Nos. 61103067 and 61273304) and the Fundamental Research Funds for the Central Universities.

References

[1] E. K. Park, D. Y. Ra, M. G. Jang, Techniques for improving web retrieval effectiveness, Information Processing & Management 41: 5 (2005) 1207-1223.

[2] W. Y. Liu, T. Y. Hao, W. Chen, et al., A web-based platform for user-interactive question-answering, World Wide Web 12: 2 (2009) 107-124.

[3] M. Donald, D. Susan, M. Christopher, Similarity measures for short segments of text, Lecture Notes in Computer Science 4425 (2007) 16-27.

[4] J. Tang, X. Wang, H. Gao, et al., Enriching short text representation in microblog for clustering, Frontiers of Computer Science in China 6: 1 (2012) 88-101.

[5] G. Salton, A. Wong, C. S. Yang, A vector space model for automatic indexing, Communications of the ACM 18: 11 (1975) 613-620.

[6] X. H. Phan, M. L. Nguyen, S. Horiguchi, Learning to classify short and sparse text & web with hidden topics from large-scale data collections, in: Proc. of WWW'08, 2008, pp. 91-100.

[7] L. Wang, Y. Jia, W. H. Han, Instant message clustering based on extended vector space model, Lecture Notes in Computer Science 4683 (2007) 435-443.

[8] M. Sahami, T. D. Heilman, A web-based kernel function for measuring the similarity of short text snippets, in: Proc. of WWW'06, 2006, pp. 377-386.

[9] W. Yih, C. Meek, Improving similarity measures for short segments of text, in: Proc. of AAAI'07, 2007, pp. 1489-1494.

[10] Y. H. Li, D. McLean, Z. A. Bandar, et al., Sentence similarity based on semantic nets and corpus statistics, IEEE Transactions on Knowledge and Data Engineering 18: 8 (2006) 1138-1150.

[11] S. Banerjee, K. Ramanathan, A. Gupta, Clustering short texts using wikipedia, in: Proc. of SIGIR'07, 2007, pp. 787-788.

[12] D. M. Blei, A. Y. Ng, M. I. Jordan, Latent Dirichlet allocation, Journal of Machine Learning Research 3: 4-5 (2003) 993-1022.

[13] X. J. Quan, G. Liu, Z. Lu, et al., Short text similarity based on probabilistic topics, Knowledge Information System 25: 3 (2010) 473-491.

[14] G. Salon, C. S. Yang, On the specification of term values in automatic indexing, Journal of Documentation 29: 4 (1973) 351-372.

[15] T. L. Griffiths, M. Steyvers, Finding scientific topics, PNAS 101: 1 (2004) 5228-5235.

[16] K. Nigam, A. K. Mccallum, S. Thrun, Text classification from labeled and unlabeled documents using EM, Machine Learning 39: 2-3 (2000) 103-134.

[17] Z. Pawlak, Rough sets, International Journal of Computer and Information Sciences 11 (1982) 341-356.

[18] C. C. Chang, C. J. Lin, LIBSVM: a library for support vector machines, ACM Transactions on Intelligent Systems and Technology 27: 2 (2011) 1-27.