A Fast Approach for Semantic Similar Short Texts Retrieval

Yanhui Gu[1] **Zhenglu Yang**[2*] **Junsheng Zhou**[1]
Weiguang Qu[1] **Jinmao Wei**[2] **Xingtian Shi**[3]
[1]School of Computer Science and Technology, Nanjing Normal University, China
{gu,zhoujs,wgqu}@njnu.edu.cn
[2]CCCE&CS, Nankai University, Tianjin, China
{yangzl,weijm}@nankai.edu.cn
[3]SAP Labs China, Shanghai, China
{xingtian.shi}@sap.com

Abstract

Retrieving semantic similar short texts is a crucial issue to many applications, e.g., web search, ads matching, question-answer system, and so forth. Most of the traditional methods concentrate on how to improve the precision of the similarity measurement, while current real applications need to efficiently explore the top similar short texts semantically related to the query one. We address the efficiency issue in this paper by investigating the similarity strategies and incorporating them into the **FAST** framework (efficient **FrA**mework for semantic similar **S**hort **T**exts retrieval). We conduct comprehensive performance evaluation on real-life data which shows that our proposed method outperforms the state-of-the-art techniques.

1 Introduction

In this paper, we investigate the fast approach of short texts retrieval, which is important to many applications, e.g., web search, ads matching, question-answer system, etc. (Yu et al., 2016; Wang et al., 2015; Hua et al., 2015; Yang et al., 2015; Wang et al., 2010; Wei et al., 2008; Cui et al., 2005; Metzler et al., 2007; Ceccarelli et al., 2011; Radlinski et al., 2008). The setting of the problem is that users always ask for those most semantically related to their queries from a huge text collection. A common solution is applying the state-of-the-art short texts similarity measurement techniques (Islam and Inkpen, 2008; Li et al., 2006; Mihalcea et al., 2006; Sahami and Heilman, 2006; Tsatsaronis et al., 2010; Mohler et al., 2011; Wang et al., 2015), and then return the top-k ones by sorting them with regard to the similarity score. After surveying the previous approaches, we find that almost all the methods concentrate on how to improve the precision, i.e., effectiveness issue. In addition, the data collections which they conducted are rather small. However, the scale of the problem has dramatically increased and the current short texts similarity measurement techniques could not handle when the data collection size becomes enormous. In this paper, we aim to address the efficiency issue in the literature while keeping their high precision. Moreover, we focus on the top-k issue because users commonly do not care about the individual similarity score but only the sorted results. Furthermore, most of the previous studies (Islam and Inkpen, 2008; Li et al., 2006; Tsatsaronis et al., 2010; Wang et al., 2015) need to set predefined threshold to filter out those dissimilar texts which is rather difficult to determine by users.

Different from long texts, short texts cannot always observe the syntax of a written language and usually do not possess sufficient information to support statistical based text processing techniques, e.g., TF-IDF. This indicates that the traditional NLP techniques for long texts may not be always appropriate to apply to short texts. The related works on short texts similarity measurement can be classified into the following major categories, i.e., (1) inner resource based strategy (Li et al., 2006; Islam and Inkpen, 2008); (2) outer resource based strategy (Tsatsaronis et al., 2010; Mihalcea et al., 2006; Islam and Inkpen, 2008; Wang et al., 2015); and (3) hybrid based strategy (Islam and Inkpen, 2008; Li et al., 2006; Wang et al., 2015).

Naively testing the candidate short texts for top-k similar short texts retrieval is inefficient when directly using these strategies. To tackle the efficiency problem, we propose an efficient strategy

* Corresponding author.

Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pages 89–94,
Berlin, Germany, August 7-12, 2016.

to evaluate as few candidates as possible. Moreover, our fast algorithm aims to output the results progressively, i.e., the top-1 should be obtained instantly. This scheme meets the demand of the real world applications, especially for big data environment. We list our contribution of this paper as follows: we propose a fast approach to tackle the efficiency problem for retrieving top-k semantic similar short texts; we present the optimized techniques and improve the efficiency which minimizes the candidate number to be evaluated in our framework. The results of four different settings demonstrate that the efficiency of our fast approach outperforms the state-of-the-art methods while keeping effectiveness.

2 Preliminaries

Formally, for a given query short text q, retrieving a set of k short texts T_s in a data collection D_s which are most semantically similar to q, i.e., $\forall t \in T_s$ and $\forall r \in (D_s - T_s)$ will yield $sim(q,t) \geq sim(q,r)$. To obtain the similarity score $sim(q,t)$ between two short texts, we can apply the current state-of-the-art strategies (Tsatsaronis et al., 2010; Mihalcea et al., 2006; Islam and Inkpen, 2008; Wang et al., 2015). In this paper, we judiciously select some similarity metrics which are assembled into a general framework to tackle the efficiency problem. Most of the existing strategies of evaluating the similarity between short texts are based on word similarity, because of the intuitive idea that short text is composed of words. As a result, we introduce the representative word similarity in the next section.

2.1 Selected Representative Similarity Measurement Strategies

There are a number of semantic similarity strategies having been developed in the previous decades which are useful in some specific applications of NLP tasks. Recently, outer resources are indispensable for short texts similarity measurement (Tsatsaronis et al., 2010; Mihalcea et al., 2006; Islam and Inkpen, 2008; Wang et al., 2015; Hua et al., 2015). After extensively investigating a number of similarity measurement strategies, we judiciously explore two representative word similarity measurement strategies which obtain the best performance compared with human judges.

Knowledge based Strategy

Knowledge based strategy determines whether two words are semantically similar by measuring their shortest path in the predefined taxonomy. The path between them can be calculated by applying word thesauri, e.g., WordNet. In this paper, we take one representative metric which has been proposed in (Leacock and Chodorow, 1998). Let's take two words w_i,w_j as an example, the similarity is as follows:

$$Sim_k(w_i, w_j) = -ln\frac{path_s(w_i, w_j)}{2 * D}$$

where $path_s(w_i, w_j)$ is the shortest path between two word concepts by using related strategy, e.g., node-counting strategy. D is the maximum depth of such taxonomy (D is with different size in either noun taxonomy or verb taxonomy).

Corpus based Strategy

Different from knowledge based strategy, corpus based strategy cannot form a new entity which means we can only apply statistical information to determine the similarity between two words. There are a few corpus based similarity measurement strategies, e.g., PMI, LSA, HAL, and so forth. In this paper, we select a representative strategy which applies Wiki encyclopedia to map Wiki texts into appropriate topics. Each Wiki topic is represented as an attribute vector. The words in the vector occur in the corresponding articles. Entries of these vectors are assigned weight which quantifies the association between words and each Wiki topic after applying vector based scheme, e.g., TF-IDF. The similarity can be evaluated by aggregating each word distributing on these topics. In addition, a short text is a vector based on topics with weight of each topic T_i formulated as: $\sum_{w_i \in T_s} v_i \cdot d_j$, where v_i is TF-IDF weight of w_i and d_j which quantifies the degree of association of word w_i with Wiki topic T_j. Here, the Wiki topic could be concepts or topics generated by other techniques, e.g., LDA, LSA, etc.

2.2 Semantic Similarity Measurement between Two Short Texts

Semantic similarity between two short texts can be measured by combining the words similarity in a general framework. Therefore, the method of combining the words similarities into a framework may affect the efficiency and effectiveness of the similarity score. In this paper, we integrate differ-

ent similarity strategies linearly and this method has been proved that it has high precision by comparing with human judges (Li et al., 2006; Islam and Inkpen, 2008). The scheme measures each word pair of short texts and then constructs a similarity score matrix. Finally, the similarity score between two short texts is recursively executed by aggregating the representative words.

3 A Fast Approach for Semantic Similar Short Texts Retrieval

We propose a fast approach for retrieving the top-k semantic similar short texts to a given query q in this section. The key idea of this scheme is to access a rather small size of candidates in the whole data collection. The scheme is conducted by building appropriate indices in offline procedure, i.e., preprocessing procedure. We illustrate the whole framework in Figure 1. The figure tells us, to efficiently retrieve top-k similar short texts, our proposed strategy only accesses as small as possible part of candidates which are filled in grey color.

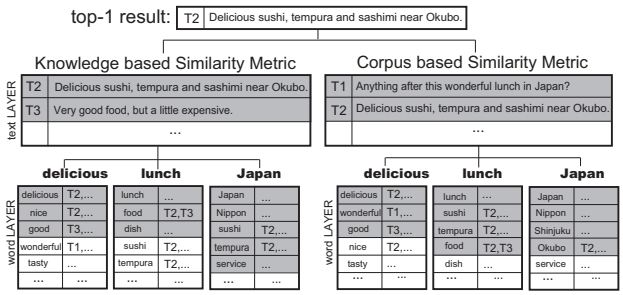

Figure 1: The framework of proposed fast approach

3.1 Efficiently Aggregate Similarity Metrics

In this section, we present an efficient assembling strategy to hasten the process of retrieving top-k similar short texts (Fagin et al., 2001). A concrete example to illustrate our proposal is presented in Figure 1. For example, let the query short text is: "Delicious lunch in Japan". After preprocessing (stemming and removing the stopwords), the query is: delicious lunch Japan. From Figure 1, we can see that there is a hierarchical structure in our framework. Suppose that if we want to retrieve top-1 short text from the whole data, the ranked list (i.e., order list) of knowledge based similarity and corpus based similarity are needed respectively. From the analysis on the property of threshold algorithm, the top-1 short text comes from these two ranked lists instantly. However, we cannot know such ranking directly because these two lists are texts layer but each list has its sub layer, i.e., word layer. In this paper, we apply two kinds of similarity metrics. Therefore, there are two assembling tasks, i.e., (1)assembling knowledge based and corpus based similarities; and (2)assembling words to texts. The words are query words and each query word corresponds to a list which can be found in Figure 1. Figure 1 also tells us for each word, it has the corresponding list in which all the words have been ranked based on the relatedness with such word. Since each word may occur in several short texts, the proposed method here should take the ID of each short text into consideration (e.g., word "delicious" occurs T2, etc.). We apply threshold algorithm to obtain the top short texts based on each query word. Therefore, the top-1 result comes from these two ranked lists based on threshold algorithm. In this example, T2 is finally outputted as the top-1 value.

3.2 Ranking list on Similarity Strategies

From the description in Section 3.1, we can see that the ranked list is crucial for using threshold algorithm to retrieve top-k short texts. In this section, we introduce the optimized method on each similarity metric.

Ranking on Knowledge based strategy
Since WordNet is a representative knowledge base, we apply the Leacock and Chodorow strategy as a WordNet evaluator which optimized as an efficient technique (Yang and Kitsuregawa, 2011).

Lemma 1 (Ordering in WordNet) *Let q be the query. Let P and S be two candidates that exist in the same taxonomy of q, that is, T_P and T_q. The shortest path between q and P (or S) is L_P in T_P (or L_S in T_S). The maximum depth of T_P is D_P (or D_S of T_S). P is more similar to Q compared with S. Thus, we have $\frac{D_P}{L_P} > \frac{D_S}{L_S}$.*

The lemma tells us that the similarity ordering between candidates in WordNet depends on the integration of the shortest path and the maximum depth of the taxonomy. We access the related synonyms set between two taxonomies successively based on the value of $\frac{D}{L}$ and obtain the top-k results in a progressive manner.

Ranking on Corpus based Strategy
We measure the similarity between short texts

by aggregating each word distribution on topics. A short text is a valued vector based on topics, where the weight of each topic T_i calculated as: $\sum_{w_i \in T_s} v_i \cdot k_j$, where v_i is TF-IDF weight of w_i and k_j which quantifies the strength of association of word w_i with Wiki topic T_j. Different from the traditional approaches, we first calculate all the similarity scores between each word in Wiki and that between topics in the data collection to obtain a set of lists during preprocessing. The topic could be generated either by ESA or by LDA. After that, we build a weighted inverted list where each list presents a word with sorted corresponding short texts according to the similarity score. Therefore, for a given query text q, each word in q corresponds to a list of short texts. As that, we apply the threshold algorithm retrieve the top-k results by using this manner. This manner accesses a small size of components of the data without necessity to evaluate every candidate short text.

After obtaining all the ranking lists, we can apply the threshold algorithm aforementioned to efficiently retrieve the top-k semantic similar short texts either by equal weight scheme or weight tuning strategy.

4 Experimental Evaluation

In this section, we conduct on three different datasets to evaluate the performance of our approach. To evaluate the effectiveness, we test the dataset which was used in (Li et al., 2006). For efficiency evaluation, we apply the BNC and MSC datasets which are extracted from British National Corpus and Microsoft Research Paraphrase Corpus respectively. The baseline strategy is implemented according to the state-of-the-art (linear assembling strategy as (Islam and Inkpen, 2008)). In our proposed strategy, we take four different settings: (1) $\mathbf{FAST}_E$ is the one that we apply the ESA topic strategy; (2) $\mathbf{FAST}_L$ employs the LDA topic strategy in corpus based similarity with equal weight; and (3) $\mathbf{FAST}_{Ew}$ and $\mathbf{FAST}_{Lw}$ are implemented based on the former two ones, respectively, with the tuned combinational weights.

4.1 Efficiency Evaluation

We evaluate the efficiency by using two real-life datasets which have been denoted as BNC and MSC. To test the effect of size of data collection, we select different size of these two datasets. Firstly, we conducted experiments on the fixed size of data collection by using 4 settings of our proposed approach. The results show that comparing with the baseline strategy, $\mathbf{FAST}_E$, $\mathbf{FAST}_L$, $\mathbf{FAST}_{Ew}$ and $\mathbf{FAST}_{Lw}$ have promotion at 75.34%, 74.68%, 75.31% and 74.59% respectively. The four settings have similar results which indicates that the weight is not the crucial factor in our proposed strategy. Table. 1 tells us the number of candidates accessed. Our evaluation has been conducted on different data collection size to test the scalability of our proposed strategy. Since the baseline strategy should access all the short texts in each size of data collection, which means in 1k size of BNC data collection, the baseline strategy access all these 1k candidates. However, our proposed strategies under different settings only access small size candidates to obtain the results. From the table, we can see that, our proposed strategy can largely reduce the number of candidates accessed in both data collections. In addition, the number of candidates accessed has increases not quickly which indicate our proposed approach scales well. Therefore, the proposed strategy is efficient than the baseline strategy.

Strategies	BNC (#Candidates accessed)			
	1k	5k	10k	20k
$\mathbf{FAST}_E$	215	1,368	1,559	1,974
$\mathbf{FAST}_L$	217	1,478	1,551	2,001
$\mathbf{FAST}_{Ew}$	225	1,511	1,621	2,043
$\mathbf{FAST}_{Lw}$	225	1,521	1,603	2,025
Strategies	**MSC (#Candidates accessed)**			
	10%	20%	50%	100%
$\mathbf{FAST}_E$	74	304	712	1,253
$\mathbf{FAST}_L$	85	313	705	1,128
$\mathbf{FAST}_{Ew}$	87	308	725	1,135
$\mathbf{FAST}_{Lw}$	81	309	712	1,076

Table 1: Number of candidates accessed in efficiency evaluation

We also evaluate the effect of k which is an important factor for evaluating the efficiency of an algorithm. The experiments conducted on a fixed size of data collection which show that the top-1 value has been outputted instantly by apply our proposed strategy while baseline strategy should access all candidates. For the query time of $\mathbf{FAST}_E$ setting costs only 19.12s while baseline strategy costs 897.5s for obtaining the top-1 value. $\mathbf{FAST}_L$, $\mathbf{FAST}_{Es}$ and $\mathbf{FAST}_{Lw}$ cost 20.13s, 21.21s and 20.32s respectively which confirms that combinational weight is not an important factor in our proposed strategy.

4.2 Effectiveness Evaluation

We illustrate the results of the correlation coefficient with human ratings in Table. 2. Note here, the baseline strategy is composed by knowledge based strategy and corpus based strategy (ESA method) with equal weight. From the table we can see that, the $\mathbf{FAST}_E$ has the same precision as the baseline because our proposed strategy only changes the order of the evaluated short texts but not the similarity strategy. $\mathbf{FAST}_L$ has better precision than $\mathbf{FAST}_E$ because we select the best LDA topic size to form Wiki topic. $\mathbf{FAST}_{Ew}$ and $\mathbf{FAST}_{Lw}$ have dynamically changed the combinational weights and therefore, the performance of them has been improved.

Baseline	Proposed Strategies			
	$\mathbf{FAST}_E$	$\mathbf{FAST}_L$	$\mathbf{FAST}_{Ew}$	$\mathbf{FAST}_{Lw}$
0.72162	0.72162	0.73333	0.74788	0.74941

Table 2: Effectiveness evaluation on different strategies

5 Conclusion

In this paper, we propose a fast approach to tackle the efficiency problem of retrieving top-k similar short texts which has not been extensively studied before. We select two representative similarity metrics, i.e., knowledge based and corpus based similarity. Efficient strategies are introduced to test as few candidates as possible in the querying process. Four different settings have been proposed to improve the effectiveness. The comprehensive experiments demonstrate the efficiency of the proposed techniques while keeping the high precision. In the future, we will investigate new methods to tackle efficiency issue and take effect semantic similarity strategies to obtain high performance.

Acknowledgment. We would like to thank the anonymous reviewers for their insightful comments. This work is partially supported by Chinese National Fund of Natural Science under Grant 61272221, 61472191, 61070089, 11431006, Jiangsu Province Fund of Social Science under Grant 12YYA002, the Natural Science Research of Jiangsu Higher Education Institutions of China under Grant 14KJB520022, and the Science Foundation of TianJin under grant 14JCYBJC15700.

References

Diego Ceccarelli, Claudio Lucchese, Salvatore Orlando, Raffaele Perego, and Fabrizio Silvestri. 2011. Caching query-biased snippets for efficient retrieval. In *Proceedings of the International Conference on Extending Database Technology*, EDBT/ICDT '11, pages 93–104.

Hang Cui, Renxu Sun, Keya Li, Min-Yen Kan, and Tat-Seng Chua. 2005. Question answering passage retrieval using dependency relations. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, pages 400–407.

Ronald Fagin, Amnon Lotem, and Moni Naor. 2001. Optimal aggregation algorithms for middleware. In *Proceedings of the ACM SIGMOD symposium on Principles of Database Systems*, PODS '01, pages 102–113.

Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. 2015. Short text understanding through lexical-semantic analysis. In *31st IEEE International Conference on Data Engineering*, ICDE'15, pages 495–506.

Aminul Islam and Diana Inkpen. 2008. Semantic text similarity using corpus-based word similarity and string similarity. *ACM Transactions on Knowledge Discovery from Data*, 2(2):1–25.

C. Leacock and M. Chodorow. 1998. Combining local context and wordnet similarity for word sense identification. In *WordNet: An Electronic Lexical Database*, pages 305–332. In C. Fellbaum (Ed.), MIT Press.

Yuhua Li, David McLean, Zuhair Bandar, James O'Shea, and Keeley A. Crockett. 2006. Sentence similarity based on semantic nets and corpus statistics. *IEEE Transactions on Knowledge and Data Engineering*, 18(8):1138–1150.

Donald Metzler, Susan T. Dumais, and Christopher Meek. 2007. Similarity measures for short segments of text. In *Proceedings of the European Conference on Information Retrieval*, ECIR '07, pages 16–27.

Rada Mihalcea, Courtney Corley, and Carlo Strapparava. 2006. Corpus-based and knowledge-based measures of text semantic similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence*, AAAI'06, pages 775–780.

Michael Mohler, Razvan C. Bunescu, and Rada Mihalcea. 2011. Learning to grade short answer questions using semantic similarity measures and dependency graph alignments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference*, ACL'11, pages 752–762.

Filip Radlinski, Andrei Broder, Peter Ciccolo, Evgeniy Gabrilovich, Vanja Josifovski, and Lance Riedel. 2008. Optimizing relevance and revenue in ad search: a query substitution approach. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 403–410.

Mehran Sahami and Timothy D. Heilman. 2006. A web-based kernel function for measuring the similarity of short text snippets. In *Proceedings of the International Conference on World Wide Web*, WWW '06.

George Tsatsaronis, Iraklis Varlamis, and Michalis Vazirgiannis. 2010. Text relatedness based on a word thesaurus. *Journal of Artificial Intelligence Research*, 37:1–39.

Kai Wang, Zhao-Yan Ming, Xia Hu, and Tat-Seng Chua. 2010. Segmentation of multi-sentence questions: towards effective question retrieval in cqa services. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '10, pages 387–394.

Peng Wang, Jiaming Xu, Bo Xu, Cheng-Lin Liu, Heng Zhang, Fangyuan Wang, and Hongwei Hao. 2015. Semantic clustering and convolutional neural network for short text categorization. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing*, ACL '15, pages 352–357.

Furu Wei, Wenjie Li, Qin Lu, and Yanxiang He. 2008. Query-sensitive mutual reinforcement chain and its application in query-oriented multi-document summarization. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 283–290.

Zhenglu Yang and Masaru Kitsuregawa. 2011. Efficient searching top-k semantic similar words. In *Proceedings of the International Joint Conference on Artificial Intelligence*, IJCAI'11, pages 2373–2378.

Shansong Yang, Weiming Lu, Dezhi Yang, Liang Yao, and Baogang Wei. 2015. Short text understanding by leveraging knowledge into topic model. In *The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL/HLT'15, pages 1232–1237.

Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2016. Understanding short texts through semantic enrichment and hashing. *IEEE Trans. Knowl. Data Eng.*, 28(2):566–579.