Automatic Multi-document Summarization Based on New Sentence Similarity Measures

Wenpeng Yin, Yulong Pei, and Lian'en Huang

Shenzhen Key Lab for Cloud Computing Technology and Applications
Peking University Shenzhen Graduate School
Shenzhen, Guangdong 518055, P.R. China
mr.yinwenpeng@gmail.com, peiyulong@sz.pku.edu.cn, hle@net.pku.edu.cn

Abstract. The acquiring of sentence similarity has become a crucial step in graph-based multi-document summarization algorithms which have been intensively studied during the past decade. Previous algorithms generally considered sentence-level structure information and semantic similarity separately, which, consequently, had no access to grab similarity information comprehensively. In this paper, we present a general framework to exemplify how to combine the two factors above together so as to derive a corpus-oriented and more discriminative sentence similarity. Experimental results on the DUC2004 dataset demonstrate that our approaches could improve the multi-document summarization performance to a considerable extent.

Keywords: graph-based multi-document summarization, sentence similarity, LDA.

1 Introduction

Sentence-based extractive summarization is a typical category of automatic document summarization and it is commonly on the basis of graph-based ranking algorithms, such as TextRank [7]. Usually, such ranking approaches use some kinds of similarity metrics to rank sentences for inclusion in the summary. The similarity of sentences can be determined by many means which can be roughly comprehended in two levels: word space based level and semantic space based level. However, the former is somewhat strict and inflexible because it depends on hard matching of terms, in which case, synonyms, hypernyms and hyponyms are treated thoroughly differently even though a term is supposed to share some similar treatments with its relatives. The other extreme is that the semantic level places too much emphasis on semantic relationship between sentences, which results in losing sentence-level structure similarity that could play as an important indicator in differentiating sentences while measuring similarity.

In order to improve the quality of summary produced via graph-based summarization algorithm, we present a framework combining sentence-level structure similarity and semantic similarity together to address the limitations of existing approaches in deriving sentence similarity. Lin et al. [5] described three methods

P. Anthony, M. Ishizuka, and D. Lukose (Eds.): PRICAI 2012, LNAI 7458, pp. 832–837, 2012.
© Springer-Verlag Berlin Heidelberg 2012

to measure sentence similarity based on term order information: longest common subsequence (LCS), weighted longest common subsequence (WLCS) and skip-bigram co-occurrence statistics, all of which could reveal the sentence-level structure similarity very well. When it turns to semantic aspect, Latent Dirichlet Allocation (LDA) [2], a latent topic model, is an appropriate tool to measure word similarity because it could capture the patterns of word usage by analyzing its context. Thus, we combine LDA with LCS, WLCS and skip-bigram respectively to design three soft matching algorithms to illuminate our intention. One advantage of our approaches is that they consider lexical order information as well as semantic relationship. The other advantage lies in the ability to identify the different senses of words with respect to their co-occurring context and consequently acquire the similarity variably. Experiments on the DUC2004 corpus demonstrate the good effectiveness of the proposed algorithms in promoting multi-document summarization performance.

2 Related Work

Since this work focuses on proposing new sentence similarity measures for graph-based summarization algorithm so as to improve the system performance, we briefly introduce some summarization methods relevant to sentence similarity and some representative approaches measuring sentence relatedness.

Famous graph-based ranking algorithms TextRank [7] and LexPageRank [3] have been successfully applied to document summarization domain, they conduct PageRank algorithm on a weighted graph, where the vertices are sentences and the weighted edge indicates the relevance of two sentences, which is acquired by using cosine measure. The task in [1] presented a method to measure dissimilarity between sentences using the normalized google distance, then performed sentence clustering for automatic text summarization.

Zhang et al. [9] indicated that sentence similarities based on word set and word order have better performance than other sentence similarities. Sentence similarity based on TF-IDF has lower precision rate, recall rate and F-measure. The work in [4] is similar to ours. It presented an algorithm that took account of semantic information and word order information implied in the sentences. The semantic similarity of two sentences was calculated using information from a structured lexical database and from corpus statistics. Word order similarity was determined by the normalized difference of word order.

3 A New Word Similarity Algorithm Based on LDA

The ability capturing semantic relations between words of Latent Dirichlet Allocation (LDA) [2] is achieved by exploiting word co-occurrence: words which co-occur in the same contexts are projected onto the same latent topic, and words that occur in different contexts are projected onto different latent topics. That's to say, words with same latent topic are supposed to possess a certain degree of similarity in semantic respect. In this study, we propose that terms

assigned same latent topic have a similarity value ranging from 0 to 1 and the concrete value could be determined by calculating the Kullback-Leibler(KL) Divergence of their distributions over latent topics. According to the Bayes rule, the probability of a specific topic z_k given a word w_v in the documents D is:

$$P(z_k|w_v, D) = \frac{P(w_v|z_k) \cdot P(z_k|D)}{P(w_v|D)} \quad . \tag{1}$$

Then the divergence of two terms w_i (probability distribution P_{w_i}) and w_j (probability distribution P_{w_j}) is determined as follows:

$$D\left(w_i, w_j\right) = KL\left(P_{w_i}, P_{w_j}\right) + KL\left(P_{w_j}, P_{w_i}\right) \quad . \tag{2}$$

Since KL divergence is asymmetric, we apply the above KL divergence-based symmetric measure. The divergence is transformed into similarity measure [6]:

$$Simi\left(w_i, w_j\right) = 10^{-\delta D(w_i, w_j)} \quad . \tag{3}$$

In experiments, we use the GibbsLDA++[1], a C/C++ implementation of LDA using Gibbs Sampling.

4 Sentence Similarity Measures

4.1 LCS_LDA

In our modified LCS (hereafter LCS_LDA), given the following original sentences:

$$S1 : \text{boy}_1\ \text{enjoy}_2\ \text{happy}_3\ \textit{holiday}_7 \quad S2 : \text{boy}_1\ \text{enjoy}_2\ \text{happy}_3\ \textit{vacation}_7$$

The subscript denotes the topic index assigned to the corresponding word. We could easily derive that the traditional LCS of S1 and S2 is 3 (hereafter, we use LCS on behalf of the length of LCS directly, such principle also applies to WLCS and Skip-Bigram cases). Instead, in our new scenario, we consider their topic sequences firstly. Consequently, the LCS of S1 and S2 seems to be 4. However, the rationale of our method lies in that although the topic indexes of two words in two different sentences are the same, the similarity value of the two word strings depends on LDA. For instance, assume that in Equation 3, "holiday" and "vacation" own a similarity value 0.9, in other words, the LCS_LDA of S1 and S2 has changed to be 3.9 in our proposed algorithm. Undoubtedly, 3.9 could reflect the length of *longest approximate subsequence* between S1 and S2 more exactly than 3 obtained in traditional LCS algorithm.

In general, the LCS_LDA score of two sentences could be computed using a analogous algorithm with LCS in [5], the key difference lies in that the variation of score in each step during the entire computing process is more likely a decimal based on Equation 3 rather than an integer 1. Therefore, inspired by [5], given the LCS_LDA score (LL for convenience) of two sentences X of length m and Y

[1] GibbsLDA++: `http://gibbslda.sourceforge.net`

of length n, we could derive the their similarity $Simi_{LL}(X, Y)$ using the following equations:

$$R_{LL} = \frac{LL(X,Y)}{m} \quad P_{LL} = \frac{LL(X,Y)}{n}$$
$$Simi_{LL}(X,Y) = \frac{(1+\beta^2)R_{LL} \cdot P_{LL}}{R_{LL} + \beta^2 P_{LL}} \ , \tag{4}$$

where $\beta = P_{LL}/R_{LL}$.

4.2 WLCS_LDA

As [5] indicated, while LCS has many good properties, it does not differentiate LCSes of different spatial relations within their embedding sequences. To improve the basic LCS method, $f(\cdot)$, a function of consecutive matches, is adopted to assign different credits to consecutive in-sequence matches, which is called Weighted LCS (WLCS). Similarly, we integrate LDA with WLCS based on the similar principle in LCS_LDA. Given the WLCS_LDA score (WL for convenience) of two sentences X of length m and Y of length n, their similarity $Simi_{WL}(X, Y)$ could be derived using the following equations:

$$R_{WL} = f^{-1}\left(\frac{WL(X,Y)}{f(m)}\right) \quad P_{WL} = f^{-1}\left(\frac{WL(X,Y)}{f(n)}\right)$$
$$Simi_{WL}(X,Y) = \frac{(1+\beta^2)R_{WL} \cdot P_{WL}}{R_{WL} + \beta^2 P_{WL}} \ , \tag{5}$$

where $\beta = P_{WL}/R_{WL}$ and $f(k) = k^2$.

4.3 Skip-Bigram_LDA

In this section, we firstly redefine skip-bigram match as a soft one (SM) rather than a strict co-occurrence as follow:

$$SM[(t_1,t_2),(t_3,t_4)] = \begin{cases} \frac{Simi(w_1,w_3)+Simi(w_2,w_4)}{2} & \text{if } t_1 = t_3 \text{ and } t_2 = t_4 \\ 0 & \text{otherwise} \end{cases} \ , \tag{6}$$

where (t_1, t_2) and (t_3, t_4) are two topic skip-bigrams. w_1, w_2, w_3, and w_4 are word strings to which t_1, t_2, t_3 and t_4 correspond, respectively. Word similarity values $Simi(w_1, w_3)$ and $Simi(w_2, w_4)$ are computed using Equation 3.

Consider the example in Section 4.1 again, topic skip-bigram (1, 7) exists in both $S1_{topic}$ and $S2_{topic}$, $SKIP2(S1, S2)$ should increase 1 according to traditional Skip-Bigram co-ocurrence, whereas, in our soft match algorithm, the match degree between $(1, 7)_{S1}$ and $(1, 7)_{S2}$ is the average value between *Simi(employee, employee)* and *Simi(holiday, vacation)*. Therefore, say *Simi(employee, employee)* = 1 and *Simi(holiday, vacation)* = 0.96, then SM($(1, 7)_{S1}$, $(1, 7)_{S2}$)=0.98. Consequently, $SKIP2(S1, S2)$ should be merely

added to 0.98. Hereafter, we use $SKIP2_{LDA}(S1, S2)$ to represent the sum of the SM values which always result from the optimal matching between topic pairs of S1 and S2. Actually, any topic pair in a sentence is likely to match more than one topic pair of the other sentence, in such case, it bears close analysis to take the optimal pair match into account.

Given two sentences X of length m and Y of length n, the Skip-Bigram_LDA (SBL) similarity $Simi_{SBL}(X, Y)$ can be derived using the following equations:

$$R_{SBL} = \frac{SKIP2_{LDA}(X,Y)}{C(m,2)} \quad P_{SBL} = \frac{SKIP2_{LDA}(X,Y)}{C(n,2)}$$
$$Simi_{SBL}(X,Y) = \frac{(1+\beta^2)R_{SBL} \cdot P_{SBL}}{R_{SBL} + \beta^2 P_{SBL}} \quad , \tag{7}$$

where $\beta = P_{SBL}/R_{SBL}$ and $C(\cdot, \cdot)$ represents the combination calculation.

5 Experiments

5.1 Data Set and Evaluation Metric

We conduct experiments on DUC2004[2] benchmark dataset. It provides 50 document sets. According to the task definitions, systems are required to produce a concise summary for each document set and the length of summaries is limited to 665 bytes. We use the ROUGE 1.5.5[3] toolkit for evaluation, which is officially adopted by DUC for evaluating automatic generated summaries.

Documents are pre-processed by segmenting sentences and splitting words. Stop words are removed and the remaining words are stemmed using Porter stemmer[4]. Then, we utilize sentence similarity discussed in Section 4 to construct undirected weighted graphs based on the algorithm proposed in [3] for scoring and ranking all the sentences. A modified version of the MMR algorithm [8] is used to remove redundancy and choose both informative and novel sentences into the summary. In experiments we set the parameters empirically. The damping factor λ in graph algorithm is set to 0.85. The penalty degree factor ω in the modified MMR is set to 0.4. Besides, we set the parameter δ in Equation 3 to 1 and the topic number in LDA is 50.

5.2 Performance Evaluation and Comparison

In experiments we compare our improved measures with three basic methods (LCS, WLCS and Skip-Bigram) and LexPageRank which is a PageRank-based summarization algorithm on the basis of cosine similarity measure taking into account only the term co-occurrence rather than the order of words. Table 1 shows the comparison results. Seen from Table 1, LCS, WLCS and Skip-Bigram

[2] Refer to `http://www-nlpir.nist.gov/projects/duc/data.html` for a detailed description of the dataset.

[3] `http://www.isi.edu/licensed-sw/see/rouge/`

[4] Porter stemmer:`http://tartarus.org/martin/PorterStemmer/`

Table 1. Comparison results on DUC2004

Systems	ROUGE-1	ROUGE-2	ROUGE-SU4
LexPageRank	0.37875	0.08354	0.12770
LCS	0.35404	0.06521	0.11019
WLCS	0.35332	0.06987	0.11175
Skip-Bigram	0.36540	0.07764	0.11950
LCS_LDA	0.38101	0.08466	0.12989
WLCS_LDA	0.38161	0.08858	0.12993
Skip-Bigram_LDA	**0.38523**	**0.09109**	**0.13123**

all have poor performances compared with LexPagaRank, which might result from that although LexPageRank ignores the order information, LCS, WLCS and Skip-Bigram neglect some words that co-occur in two sentences while not in the common subsequence. Nevertheless, their modified versions could considerably improve the evaluation results over all three metrics, which demonstrates that combining word semantic similarity and sentence structure information does benefit the calculating of sentence semantic similarity.

Acknowledgments. This research is financially supported by NSFC of China (Grant No. 60933004 and 61103027) and HGJ (Grant No. 2011ZX01042-001-001). We thank the anonymous reviewers for their useful comments.

References

1. Aliguliyev, R.M.: A new sentence similarity measure and sentence based extractive technique for automatic text summarization. Expert Systems with Applications 36(4), 7764–7772 (2009)
2. Blei, D.M., Ng, A.Y., Jordan, M.I.: Latent dirichlet allocation. The Journal of Machine Learning Research 3, 993–1022 (2003)
3. Erkan, G., Radev, D.R.: Lexpagerank: Prestige in multi-document text summarization. In: Proceedings of EMNLP, pp. 365–371 (2004)
4. Li, Y., McLean, D., Bandar, Z.A., O'Shea, J.D., Crockett, K.: Sentence similarity based on semantic nets and corpus statistics. IEEE Transactions on Knowledge and Data Engineering 18(8), 1138–1150 (2006)
5. Lin, C., Och, F.: Automatic evaluation of machine translation quality using longest common subsequence and skip-bigram statistics. In: Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics, p. 605. Association for Computational Linguistics (2004)
6. Manning, C., Schutze, H.: Foundations of statistical natural language processing. In: Enhancing Semantic Distances With Context Awareness (1999)
7. Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proceedings of EMNLP, pp. 404–411. ACL, Barcelona (2004)
8. Wan, X.: Document-Based HITS Model for Multi-document Summarization. In: Ho, T.-B., Zhou, Z.-H. (eds.) PRICAI 2008. LNCS (LNAI), vol. 5351, pp. 454–465. Springer, Heidelberg (2008)
9. Zhang, J., Sun, Y., Wang, H., He, Y.: Calculating statistical similarity between sentences. Journal of Convergence Information Technology 6(2) (2011)