Text Classification based on the Latent Topics of Important Sentences extracted by the PageRank Algorithm

Yukari Ogura and Ichiro Kobayashi
Advanced Sciences, Graduate School of Humanities and Sciences,
Ochanomizu University
2-1-1 Ohtsuka Bunkyo-ku Tokyo, 112-8610 JAPAN
{ogura.yukari, koba}@is.ocha.ac.jp

Abstract

In this paper, we propose a method to raise the accuracy of text classification based on latent topics, reconsidering the techniques necessary for good classification – for example, to decide important sentences in a document, the sentences with important words are usually regarded as important sentences. In this case, *tf.idf* is often used to decide important words. On the other hand, we apply the PageRank algorithm to rank important words in each document. Furthermore, before clustering documents, we refine the target documents by representing them as a collection of important sentences in each document. We then classify the documents based on latent information in the documents. As a clustering method, we employ the k-means algorithm and investigate how our proposed method works for good clustering. We conduct experiments with Reuters-21578 corpus under various conditions of important sentence extraction, using latent and surface information for clustering, and have confirmed that our proposed method provides better result among various conditions for clustering.

1 Introduction

Text classification is an essential issue in the field of natural language processing and many techniques using latent topics have so far been proposed and used under many purposes. In this paper, we aim to raise the accuracy of text classification using latent information by reconsidering elemental techniques necessary for good classification in the following three points: 1) important words extraction — to decide important words in documents is a crucial issue for text classification, *tf.idf* is often used to decide them. Whereas, we apply the PageRank algorithm (Brin et al., 1998) for the issue, because the algorithm scores the centrality of a node in a graph, and important words should be regarded as having the centrality (Hassan et al., 2007). Besides, the algorithm can detect centrality in any kind of graph, so we can find important words for any purposes. In our study, we express the relation of word co-occurrence in the form of a graph. This is because we use latent information to classify documents, and documents with high topic coherence tend to have high PMI of words in the documents (Newman et al., 2010). So, we construct a graph from a viewpoint of text classification based on latent topics. 2) Refinement of the original documents — we recompile the original documents with a collection of the extracted important sentences in order to refine the original documents for more sensitive to be classified. 3) Information used for classification — we use latent information estimated by latent Dirichlet allocation (LDA) (Blei et al., 2003) to classify documents, and compare the results of the cases using both surface and latent information. We experiment text classification with Reuters-21578 corpus; evaluate the result of our method with the results of those which have various other settings for classification; and show the usefulness of our proposed method.

2 Related studies

Many studies have proposed to improve the accuracy of text classification. In particular, in terms of improving a way of weighting terms in a docu-

Proceedings of the ACL Student Research Workshop, pages 46–51,
Sofia, Bulgaria, August 4-9 2013. ©2013 Association for Computational Linguistics

ment for text classification, there are many studies which use the PageRank algorithm. In (Hassan et al., 2007), they have applied a random-walk model on a graph constructed based on the words which co-occur within a given window size, e.g., 2,4,6,8 words in their experiments, and confirmed that the windows of size 2 and 4 supplied the most significant results across the multiple data set they used. Zaiane et al. (2002) and Wang et al. (2005) have introduced association rule mining to decide important words for text classification. In particular, Wang et al. have used a PageRank-style algorithm to rank words and shown their method is useful for text classification. Scheible et al. (2012) have proposed a method for bootstrapping a sentiment classifier from a seed lexicon. They apply topic-specific PageRank to a graph of both words and documents, and introduce Polarity PageRank, a new semi-supervised sentiment classifier that integrates lexicon induction with document classification. As a study related to topic detection by important words obtained by the PageRank algorithm, Kubek et al. (2011) has detected topics in a document by constructing a graph of word co-occurrence and applied the PageRank algorithm on it.

To weight words is not the issue for only text classification, but also an important issue for text summarization, Erkan et al. (2004) and Mihlcea et al. (2004b; 2004a) have proposed multi-document summarization methods using the PageRank algorithm, called LexRank and TextRank, respectively. They use PageRank scores to extract sentences which have centrality among other sentences for generating a summary from multi-documents.

On the other hand, since our method is to classify texts based on latent information. The graph used in our method is constructed based on word co-occurrence so that important words which are sensitive to latent information can be extracted by the PageRank algorithm. At this point, our attempt differs from the other approaches.

3 Techniques for text classification

3.1 Extraction of important words

To decide important words, *tf.idf* is often adopted, whereas, another methods expressing various relation among words in a form of a graph have been proposed (2005; Hassan et al., 2007). In particular, (Hassan et al., 2007) shows that the PageRank score is more clear to rank important words rather than *tf.idf*. In this study, we refer to their method and use PageRank algorithm to decide important words.

The PageRank algorithm was developed by (Brin et al., 1998). The algorithm has been used as the basic algorithm of Google search engine, and also used for many application to rank target information based on the centrality of information represented in the form of a graph.

In this study, the important words are selected based on PageRank score of a graph which represents the relation among words. In other words, in order to obtain good important sentences for classification, it is of crucial to have a good graph (Zhu et al., 2005) because the result will be considerably changed depending on what kind of a graph we will have for important words. In this study, since we use latent information for text classification, therefore, we construct a graph representing the relation of words from a viewpoint topic coherence. According to (Newman et al., 2010), topic coherence is related to word co-occurrence. Referring to their idea, we construct a graph over words in the following manner: each word is a node in the graph, and there is an undirected edge between every pair of words that appear within a three-sentence window – to take account of contextual information for words, we set a three-sentence window. We then apply the PageRank algorithm to this graph to obtain a score for every word which is a measurement of its centrality – the centrality of a word corresponds to the importance of a word. A small portion of a graph might look like the graph in Figure 1.

3.2 Refinement of target documents

After selecting important words, the important sentences are extracted until a predefined ratio of whole sentences in each document based on the selected important words, and then we reproduce refined documents with a collection of extracted important sentences. An important sentence is decided by how many important words are included in the sentence. The refined documents are composed of the important sentences extracted from a viewpoint of latent information, i.e., word co-occurrence, so they are proper to be classified based on latent information.

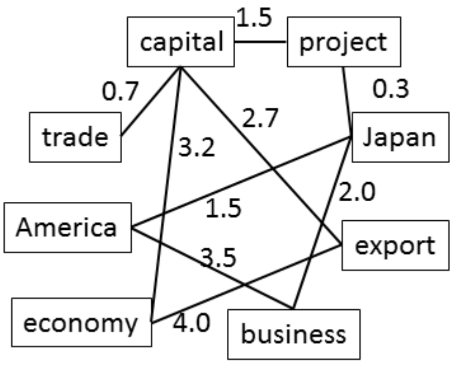


Figure 1: A graph of word cooccurrence

3.3 Clustering based on latent topics

After obtaining a collection of refined documents for classification, we adopt LDA to estimate the latent topic probabilistic distributions over the target documents and use them for clustering. In this study, we use the topic probability distribution over documents to make a topic vector for each document, and then calculate the similarity among documents.

3.4 Clustering algorithm

step.1 *Important words determination*

The important words are decided based on *tf.idf* or PageRank scores. As for the words decided based on PageRank scores, we firstly have to make a graph on which the PargeRank algorithm is applied. In our study, we construct a graph based on word co-occurrence. So, important words are selected based on the words which have centrality in terms of word co-occurrence. In particular, in our study we select co-occurred words in each three sentences in a document, taking account of the influence of contextual information.

step.2 *Refinement of the target documents*

After selecting the important words, we select the sentences with at least one of the words within the top 3 PageRank score as important sentences in each document, and then we reproduce refined documents with a collection of the extracted important sentences.

step.3 *Clustering based on latent topics*

As for the refined document obtained in step 2, the latent topics are estimated by means of LDA. Here, we decide the number of latent topics k in the target documents by measuring the value of perplexity $P(\boldsymbol{w})$ shown in equation (1). The similarity of documents are measured by the Jenshen-Shannon divergence shown in equation (2).

$$P(\boldsymbol{w}) = exp(-\frac{1}{N}\sum_{mn} log(\sum_{z} \theta_{mz}\phi_{zw_{mn}})) \tag{1}$$

Here, N is the number of all words in the target documents, w_{mn} is the n-th word in the m-th document; θ is the topic probabilistic distribution for the documents, and ϕ is the word probabilistic distribution for every topic.

$$\begin{aligned} & D_{JS}(P||Q) \\ & = \frac{1}{2}(\sum_{x} P(x)log\frac{P(x)}{R(x)} + \sum_{x} log\frac{Q(x)}{R(x)}) \\ & \qquad where, R(x) = \frac{P(x)+Q(x)}{2} \end{aligned} \tag{2}$$

4 Experiment

We evaluate our proposed method by comparing the accuracy of document clustering between our method and the method using *tf.idf* for extracting important words.

4.1 Experimental settings

As the documents for experiments, we use Reuters-21578 dataset [1] collected from the Reuters newswire in 1987.In our proposed method, the refined documents consisting of important sentences extracted from the original documents are classified, therefore, if there are not many sentences in a document, we will not be able to verify the usefulness of our proposed method. So, we use the documents which have more than 5 sentences in themselves. Of the 135 potential topic categories in Reuters-21578, referring to other clustering study (Erkan, 2006; 2005; Subramanya et al., 2008), we also use the most frequent 10 categories: i.e., *earn, acq, grain, wheat, money, crude, trade, interest, ship, corn*. In the

[1] http://www.daviddlewis.com/resources/testcollections/reuters21578/

sequel, we use 792 documents whose number of words is 15,835 for experiments – the 792 documents are the all documents which have more than 5 sentences in themselves in the corpus. For each document, stemming and stop-word removal processes are adopted. Furthermore, the hyper-parameters for topic probability distribution and word probability distribution in LDA are α=0.5 and β=0.5, respectively. We use Gibbs sampling and the number of iteration is 200. The number of latent topics is decided by perplexity, and we decide the optimal number of topics by the minimum value of the average of 10 times trial, changing the number of topics ranging from 1 to 30.

As the first step for clustering with our method, in this study we employ the k-means clustering algorithm because it is a representative and a simple clustering algorithm.

4.2 Evaluation method

For evaluation, we use both accuracy and F-value, referring to the methods used in (Erkan, 2006). As for a document d_i, l_i is the label provided to d_i by the clustering algorithm, and α_i is the correct label for d_i. The accuracy is expressed in equation (3).

$$Accuracy = \frac{\sum_{i=1}^{n} \delta\left(map\left(l_i\right), \alpha_i\right)}{n} \tag{3}$$

$\delta\left(x, y\right)$ is 1 if $x = y$, otherwise 0. $map\left(l_i\right)$ is the label provided to d_i by the k-means clustering algorithm. For evaluation, the F-value of each category is computed and then the average of the F-values of the whole categories, used as an index for evaluation, is computed (see, equation (4)).

$$F = \frac{1}{|C|} \sum_{c_i \in C} F\left(c_i\right) \tag{4}$$

As the initial data for the k-means clustering algorithm, a correct document of each category is randomly selected and provided. By this, the category of classified data can be identified as in (Erkan, 2006).

4.3 Experiment results

To obtain the final result of the experiment, we applied the k-means clustering algorithm for 10 times for the data set and averaged the results. Here, in the case of clustering the documents based on the topic probabilistic distribution by LDA, the topic distribution over documents θ is changed in every estimation. Therefore, we estimated θ for 8 times and then applied the k-means clustering algorithm with each θ for 10 times. We averaged the results of the 10 trials and finally evaluated it. The number of latent topics was estimated as 11 by perplexity. We used it in the experiments. To measure the latent similarity among documents, we construct topic vectors with the topic probabilistic distribution, and then adopt the Jensen-Shannon divergence to measures it, on the other hand, in the case of using document vectors we adopt cosine similarity.

Table 1 and Table 2 show the cases of with and without refining the original documents by recompiling the original documents with the important sentences.

Table 1: Extracting important sentences

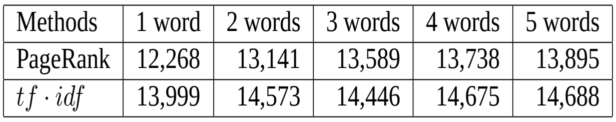
Methods	Measure	Accuracy	F-value
PageRank	Jenshen-Shannon	**0.567**	**0.485**
	Cosine similarity	0.287	0.291
tf.idf	Jenshen-Shannon	0.550	0.435
	Cosine similarity	0.275	0.270

Table 2: Without extracting important sentences

Similarity measure	Accuracy	F-value
Jenshen-Shannon	0.518	0.426
Cosine similarity	0.288	0.305

Table 3, 4 show the number of words and sentences after applying each method to decide important words.

Table 3: Change of number of words



Methods	1 word	2 words	3 words	4 words	5 words
PageRank	12,268	13,141	13,589	13,738	13,895
$tf \cdot idf$	13,999	14,573	14,446	14,675	14,688

Furthermore, Table 5 and 6 show the accuracy and F-value of both methods, i.e., PageRank scores and *tf.idf*, in the case that we use the same number of sentences in the experiment to experiment under the same conditions.

Table 4: Change of number of sentences

Methods	1 word	2 words	3 words	4 words	5 words
PageRank	1,244	1,392	1,470	1,512	1,535
$tf \cdot idf$	1,462	1,586	1,621	1,643	1,647

Table 5: Accuracy to the number of topics

Num. of topics	8	9	10	11	12
PageRank	0.525	0.535	**0.566**	0.553	0.524
tf.idf	0.556	0.525	0.557	0.550	0.541

4.4 Discussion

We see from the experiment results that as for the measures based on the Jenshen-Shannon divergence, both accuracy and F-value of the case where refined documents are clustered is better than the case where the original documents are clustered. We have conducted t-test to confirm whether or not there is significant difference between the cases: with and without extracting important sentences. As a result, there is significant difference with 5 % and 1 % level for the accuracy and F-value, respectively.

When extracting important sentences, although the size of the document set to be clustered is smaller than the original set, the accuracy increases. So, it can be said that necessary information for clustering is adequately extracted from the original document set.

From this, we have confirmed that the documents are well refined for better clustering by recompiling the documents with important sentences. We think the reason for this is because only important sentences representing the contents of a document are remained by refining the original documents and then it would become easier to measure the difference between probabilistic distributions of topics in a document. Moreover, as for extracting important sentences, we confirmed that the accuracy of the case of using PageRank scores is better than the case of using *tf.idf*. By this, constructing a graph based on word co-occurrence of each 3 sentences in a document works well to rank important words, taking account of the context of the word.

We see from Table 3 , 4 that the number of words and sentences decreases when applying PageRank scores. In the case of applying *tf.idf*, the *tf.idf* value tends to be higher for the words which often appear in a particular document. Therefore, the extraction of sentences including the words with high *tf.idf* value may naturally lead to the extraction of many sentences.

Table 6: F-value to the number of topics

Num. of topics	8	9	10	11	12
PageRank	0.431	0.431	**0.467**	0.460	0.434
tf.idf	0.466	0.430	0.461	0.435	0.445

The reason for low accuracy in the case of using cosine similarity for clustering is that it was observed that the range of similarity between documents is small, therefore, the identification of different categorized documents was not well achieved.

Table 5 and Table 6 show the accuracy and F-value to the number of latent topics, respectively. We see that both accuracy and F-value of the case of using PageRank scores are better than those of the case of using *tf.idf* in the case of the number of topics is 9,10,and 11. In particular, the highest score is made when the number of topics is 10 for both evaluation measures — we think the reason for this is because we used document sets of 10 categories, therefore, it is natural to make the highest score when the number of topics is 10. So, we had better look at the score of the case where the number of topics is 10 to compare the ability of clustering. By the result, we can say that PageRank is better in refining the documents so as they suit to be classified based on latent information.

5 Conclusions

In this study, we have proposed a method of text clustering based on latent topics of important sentences in a document. The important sentences are extracted through important words decided by the PageRank algorithm. In order to verify the usefulness of our proposed method, we have conducted text clustering experiments with Reuters-21578 corpus under various conditions — we have adopted either PageRank scores or *tf.idf* to decide important words for important sentence extraction, and then adopted the k-means clustering algorithm for the documents recompiled with the extracted important sentences based on either latent or surface information. We see from the results of the experiments that the clustering based on latent information is generally better than that based on surface information in terms of clustering accuracy. Furthermore, deciding important words with PageRank scores is better than that with *tf.idf* in terms of clustering accuracy. Compared to the number of the extracted words in important sentences between PageRank scores and *tf.idf*, we see that the number of sentences extracted based on PageRank scores is smaller than that based on *tf.idf*, therefore, it can be thought that more context-sensitive sentences are extracted by adopting PageRank scores to decide important words.

As future work, since clustering accuracy will be changed by how many sentences are compiled in a refined document set, therefore, we will consider a more sophisticated way of selecting proper important sentences. Or, to avoid the problem of selecting sentences, we will also directly use the words extracted as important words for clustering. Moreover, at this moment, we use only k-means clustering algorithm, so we will adopt our proposed method to other various clustering methods to confirm the usefulness of our method.

References

David M. Blei and Andrew Y. Ng and Michael I. Jordan and John Lafferty. 2003. *Latent dirichlet allocation*, Journal of Machine Learning Research,

Sergey Brin and Lawrence Page. 1998. The Anatony of a Large-scale Hypertextual Web Search Engine, Computer Networks and ISDN Systems, pages. 107–117.

Gunes Erkan, 2004. *LexRank: Graph-based Lexical Centrality as Salience in Text Summarization* Journal of Artificial Intelligence Research 22, pages.457-479

Gunes Erkan. 2006. *Language Model-Based Document Clustering Using Random Walks*, Association for Computational Linguistics, pages.479–486.

Samer Hassan, Rada Mihalcea and Carmen Banea. 2007. *Random-Walk Term Weighting for Improved Text Classification*, SIGIR '07 Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval, pages.829-830.

Mario Kubek and Herwig Unger, 2011 *Topic Detection Based on the PageRank's Clustering Property*, IICS'11, pages.139-148,

Rada Mihalcea. 2004. *Graph-based Ranking Algorithms for Sentence Extraction, Applied to Text Summarization*, Proceeding ACLdemo '04 Proceedings of the ACL 2004 on Interactive poster and demonstration sessions Article No. 20.

Rada Mihalcea and Paul Tarau 2004. *TextRank: Bringing Order into Texts*, Conference on Empirical Methods in Natural Language Processing.

David Newman, Jey Han Lau, Karl Grieser, and Timothy Baldwin, 2010. *Automatic evaluation of topic coherence*, Human Language Technologies: The 2010 Annual Conference of the North Ametican Chapter of the Association for Computational Linguistics, pages. 100–108, Los Angeles.

Christian Scheible, Hinrich Shutze. 2012. *Bootstrapping Sentiment Labels For Unannotated Documents With Polarity PageRank*, Proceedings of the Eight International Conference on Language Resources and Evaluation.

Amarnag Subramanya, Jeff Bilmes. 2008. *Soft-Supervised Learning for Text Classification* Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing, pages.1090–1099, Honolulu.

Wei Wang, Diep Bich Do, and Xuemin Lin. 2005. *Term Graph Model for Text Classification*, Springer-Verlag Berlin Heidelberg 2005, pages.19–30.

Osmar R. Zaiane and Maria-luiza Antonie. 2002. *Classifying Text Documents by Associating Terms with Text Categories*, In Proc. of the Thirteenth Australasian Database Conference (ADC'02), pages.215–222,

X Zhu. 2005. *Semi-supervised learning with Graphs*, Ph.D thesis, Carnegie Mellon University.