PKUSUMSUM : A Java Platform for Multilingual Document Summarization

Jianmin Zhang, Tianming Wang and Xiaojun Wan
Institute of Computer Science and Technology, Peking University
Beijing 100871, China
{zhangjianmin2015, wangtm, wanxiaojun}@pku.edu.cn

Abstract

PKUSUMSUM is a Java platform for multilingual document summarization, and it supports multiple languages, integrates 10 automatic summarization methods, and tackles three typical summarization tasks. The summarization platform has been released and users can easily use and update it. In this paper, we make a brief description of the characteristics, the summarization methods, and the evaluation results of the platform, and also compare PKUSUMSUM with other summarization toolkits.

1 Introduction

Automatic document summarization has drawn much attention in the fields of natural language processing, information retrieval and text mining for a long time. It is very useful to help users quickly get main information from a long document or a large number of documents, and thus save users' reading time. In the past years, document summarization has become an active research area and various document summarization methods have been proposed. A well-designed and well-developed document summarization platform will greatly help both researchers and developers in this area, and more in-depth researches and real applications can be easily conducted and realized based on this platform. However, there are several major shortcomings in existing document summarization toolkits, e.g., low coverage of summarization methods, no support of multiple tasks and multiple languages, poor scalability, etc. Therefore, we aim at developing a more competitive document summarization platform in order to satisfy various kinds of research and development needs in this area.

Our summarization toolkit is called PKUSUMSUM (PKU's SUMmary of SUMmarization methods), which is a Java platform for multilingual document summarization. It is developed in Java and supports single-document, multi-document and topic-focused multi-document summarizations in multiple languages. More importantly, it covers a number of various summarization methods.

Main features of PKUSUMSUM include:

- It integrates stable and various summarization methods, and the performance is good enough.
- It supports three typical summarization tasks, including simple-document, multi-document and topic-focused multi-document summarizations.
- It supports Western languages (e.g. English) and Chinese language.
- It integrates English tokenizer, stemmer and Chinese word segmentation tools.
- The Java platform can be easily distributed on different OS platforms, like Windows, Linux and MacOS.
- It is open source and developed with modularization, so that users can add new methods and modules into the toolkit conveniently.

The above features makes PKUSUMSUM have significant advantages over existing automatic summarization tools which only partially fulfill the requirements illustrated above.

2 Summarization Methods for Different Summarization Tasks

PKUSUMSUM is a powerful Java platform for multilingual document summarization. It integrates 10 popular summarization methods, supports for multiple languages, and can tackle three typical document summarization tasks. The performance values of the methods implemented in PKUSUMSUM are very competitive. To be specific, PKUSUMSUM integrates the following 10 unsupervised summarization methods (including baselines):

Lead: This baseline method takes the first sentences one by one in the single document or the first document in the collection, where documents in the collection are assumed to be ordered by name.

Coverage: This baseline method takes the first sentence one by one from the first document to the last document in the collection.

Centroid: In centroid-based summarization (Radev et al., 2004a), a pseudo-sentence of the document called centroid is constructed. The centroid consists of words with TFIDF scores above a predefined threshold. The score of each sentence is defined by summing the scores based on different features including cosine similarity of the sentence with the centroid, position weight and cosine similarity with the first sentence. We also added an additional feature of cosine similarity between the sentence and the topic for the topic-based multi-document summarization task.

TextRank: TextRank (Mihalcea et al., 2004) builds a graph and adds each sentence as vertices, the overlap of two sentences as relations that connect sentences. Then the graph-based ranking algorithm is applied until convergence. Sentences are sorted based on their final score and a greedy algorithm is employed to impose diversity penalty on each sentence and select summary sentences.

LexPageRank: LexPageRank (Erkan et al., 2004) computes sentence importance based on the concept of eigenvector centrality in a graph representation of sentences. In this model, a connectivity matrix based on intra-sentence cosine similarity is used as the adjacency matrix of the graph representation of sentences.

ClusterCMRW: Given a document set covering a few topic themes, usually the sentences in an important theme cluster are deemed more salient than the sentences in a trivial theme cluster. The Cluster-based Conditional Markov Random Walk Model (ClusterCMRW) (Wan and Yang, 2008) makes use of the link relationships between sentences in the document set and fully leverages the cluster-level information.

ManifoldRank: The manifold-ranking method is a typical method for topic-focused multi-document summarization (Wan et al., 2007). The ranking score is obtained for each sentence in the manifold-ranking process to denote the biased information richness of the sentence. Then a greedy algorithm is employed to impose diversity penalty on each sentence.

ILP: Integer linear programming (ILP) approaches (Gillick et al., 2009) cast document summarization as a combinatorial optimization problem. An ILP model selects sentences by maximizing the sum of frequency-induced weights of bigram concepts contained in the summary. Here we use the open source tool lp_solve[1] for Java to solve the ILP problem.

Submodular: Using submodular function is a very competitive approach in multi-document summarization. It performs summarization by maximizing submodular functions under a budget constraint. The submodularity hidden in the coverage, diversity and non-redundancy can be reflected in a class of submodular functions. We use two submodular functions for document summarization tasks (Lin and Bilmes, 2010; Li at el, 2012). In particular, **Submodular1** implements the algorithm proposed in (Li at el, 2012) and uses formula (7) in the paper. **Submodular2** makes some modifications on the functions in (Lin and Bilmes, 2010).

As mentioned earlier, PKUSUMSUM can tackle three typical summarization tasks, and Table 1 shows which tasks can be solved by each method. “Yes” means that the method can solve the certain task.

For evaluating the performance of PKUSUMSUM, we use the DUC benchmark datasets. We use the DUC 2002 (Task 1) dataset for evaluating single-document summarization, the DUC 2004 (Task 2) dataset for evaluating multi-document summarization and the DUC 2006 dataset for evaluating topic-focused multi-document summarization. The ROUGE metrics (Lin and Hovy, 2003) are used to automatically evaluate the quality of produced summaries given the gold-standard reference summaries. We use the ROUGE-1.5.5 toolkit to perform the evaluation, and report the F-scores of the following metrics in the experimental results: ROUGE-1, ROUGE-2, and ROUGE-SU4. The scores of different methods in PKUSUMSUM for different tasks are shown in Tables 2-4, respectively. We can see that

[1] http://lpsolve.sourceforge.net/5.5/

Lead is hard to defeat for single-document summarization, while **Submodular1&2** and **ManifoldRank** perform well for multi-document and topic-focused summarizations, respectively.

Method	Single-document	Multi-document	Topic-focused Multi-document
Lead	Yes	Yes	Yes
Coverage	-	Yes	Yes
Centroid	Yes	Yes	Yes
TextRank	Yes	Yes	-
LexPageRank	Yes	Yes	-
ClusterCMRW	-	Yes	-
ManifoldRank	-	-	Yes
ILP	Yes	Yes	-
Submodular1	Yes	Yes	-
Submodular2	Yes	Yes	-

Table 1. The correspondence between summarization tasks and methods

Method	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.4770	0.2242	0.2407
Centroid	0.4755	0.2230	0.2389
TextRank	0.4562	0.1930	0.2155
LexPageRank	0.4502	0.1851	0.2093
ILP	0.4756	0.2214	0.2386
Submodular1	0.4592	0.1893	0.2122
Submodular2	0.4604	0.1924	0.2148

Table 2. F-scores for single-document summarization on DUC 2002 (Task 1)

Method	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.3182	0.0645	0.1023
Coverage	0.3392	0.0757	0.1152
Centroid	0.3668	0.0876	0.1268
TextRank	0.3725	0.0863	0.1272
LexPageRank	0.3607	0.0755	0.1202
ILP	0.3601	0.0743	0.1185
Submodular1	0.3841	0.0949	0.1348
Submodular2	0.3839	0.0958	0.1355
ClusterCMRW	0.3760	0.0908	0.1308

Table 3. F-scores for multi-document summarization on DUC 2004 (Task 2)

Method	ROUGE-1	ROUGE-2	ROUGE-SU4
Lead	0.3458	0.0589	0.1132
Coverage	0.3502	0.0643	0.1218
ManifoldRank	0.4028	0.0812	0.1387
Centroid	0.3578	0.0580	0.1134

Table 4. F-scores for topic-based multi-document summarization on DUC 2006

3 Availability, License, Usage and Scalability

The PKUSUMSUM toolkit has been released and the open-source software can be freely downloaded[2] and used under the GNU GPL license.

PKUSUMSUM is developed with Java. The Java platform can be easily distributed on different operating systems, like Windows, Linux and MacOS, so users who are used to different operating systems can use PKUSUMSUM with no barrier.

Both the source code and the Java executable package of PKUSUMSUM are provided. If users are not familiar with the Java source code or do not want to re-compile the code, they can use command line to run the Java package. The parameters in different summarization methods can be conveniently set by users and they all have default values.

We integrate some pre-processing or post-processing modules into the platform, like English tokenizer[3], stemmer[4] and Chinese word segmenter[5]. Other western languages are also supported. Users can easily obtain summaries for documents without extra processing.

[2] http://www.icst.pku.edu.cn/lcwm/wanxj/files/PKUSUMSUM.zip

[3] Stanford Tokenizer, http://nlp.stanford.edu/software/tokenizer.html

[4] The Porter Stemming Algorithm, http://tartarus.org/~martin/PorterStemmer/

PKUSUMSUM is developed with modularity and it is easy to add new modules to the platform. For example, we create an independent class for each data processing unit or summarization method, so users can add new classes for other methods without altering the structure of the platform.

4 Comparison with Other Toolkits

We compared PKUSUMSUM with the following typical existing automatic summarization toolkits:

MUSEEC (MUltilingual SEntence Extraction and Compression) (Litvak et al., 2016): This summarization tool implements only three extractive summarization techniques as MUSE based on a genetic algorithm (GA), POLY based on linear programming (LP), and an extension of POLY named WECOM. Although it can support multiple western languages, the three homogeneous methods it implements are not adequate and it is not easy to modify.

MEAD[6] (Radev et al., 2004b): The methods it implements are very limited and simple. It can tackle single and multi-document summarization tasks, but does not support topic-focused multi-document summarization task.

SUMMA (Horacio Saggion, 2008): It depends on the GATE platform (Cunningham et al., 2002), and only supports one method.

In addition, there are some simple tools coding by Python, such as **sumpy**, which can support four simple methods, and **summa**, which can only support TextRank. The existing systems have several of the following problems: 1) low coverage of summarization methods; 2) no support of different tasks; 3) no support of multiple languages; 4) poor scalability; 5) lack of platform independence.

5 Conclusion

We introduced the PKUSUMSUM platform for document summarization, which has been released. It has powerful ability and it supports multi-language, integrates 10 automatic summarization methods and can tackle three popular summarization tasks. In our future work, we will add more supervised summarzation methods into the platform.

Reference

Cunningham, Hamish, Diana Maynard, Kalina Bontchva and Valentin Tablan. 2002. A framework and graphical development environment for robust NLP tools and applications. *ACL*.

Erkan, Günes, and Dragomir R. Radev. 2004. LexPageRank: Prestige in Multi-Document Text Summarization. *EMNLP*.

Gillick, Dan, and Benoit Favre. 2009. A scalable global model for summarization. *Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing*.

Li, Jingxuan, Lei Li, and Tao Li. 2012. Multi-document summarization via submodularity. *Applied Intelligence* 37.3: 420-430.

Lin, Chin-Yew, and Eduard Hovy. 2003. Automatic evaluation of summaries using n-gram co-occurrence statistics. *NAACL*.

Lin, Hui, and Jeff Bilmes. 2010. Multi-document summarization via budgeted maximization of submodular functions. *HLT-NAACL*.

Litvak, Marina, Natalia Vanetik, Mark Last, and Elena Churkin. 2016. MUSEEC: A Multilingual Text Summarization Tool. *ACL*.

Mihalcea, Rada, and Paul Tarau. 2004. TextRank: Bringing order into texts. *EMNLP*.

Radev, Dragomir R., Hongyan Jing, Małgorzata Stys, Daniel Tam. 2004a. Centroid-based summarization of multiple documents. *Information Processing & Management* 40.6: 919-938.

Radev, Dragomir R., Timothy Allison, Sasha Blair-Goldensohn, et al. 2004b. MEAD-A Platform for Multidocument Multilingual Text Summarization. *LREC*.

Saggion, Horacio. 2008. A robust and adaptable summarization tool. Traitement Automatique des Langues 49.2.

Wan, Xiaojun, Jianwu Yang, and Jianguo Xiao. 2007. Manifold-Ranking Based Topic-Focused Multi-Document Summarization. *IJCAI*.

Wan, Xiaojun, and Jianwu Yang. 2008. Multi-document summarization using cluster-based link analysis. *SIGIR*.

[5] Ansj toolkit, https://github.com/NLPchina/ansj_seg

[6] http://www.summarization.com/mead/