README

1. Introduction

PKUSUMSUM (PKU's SUMmary of SUMmarization methods) is an integrated toolkit for automatic document summarization. It supports single-document, multi-document and topic-focused multi-document summarizations, and a variety of summarization methods have been implemented in the toolkit.

Users can easily use the toolkit to produce summaries for documents or document sets, and implement their own summarization methods based on the platform.

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 package of PKUSUMSUM includes the Jar package, source code in "/code" and referenced libraries in "/lib".

The correspondence between the summarization methods and the summarization tasks is shown in the following table:

Method	Single-document summarization	Multi-document summarization	Topic-based Multi-document summarization
Coverage	-	Yes	Yes
Lead	Yes	Yes	Yes
Centroid [1]	Yes	Yes	Yes
TextRank [2]	Yes	Yes	-
LexPageRank [3]	Yes	Yes	-
ILP [4]	Yes	Yes	-
Submodular1 [5]	Yes	Yes	-
Submodular2 [6]	Yes	Yes	-
ClusterCMRW [7]	-	Yes	-
ManifoldRank [8]	-	-	Yes

2. Notice

- We use **lp_solve for Java** to solve the ILP model. If you choose the ILP method to solve the problem, please configure lp_solve.
  - Copy the lp_solve dynamic libraries from the archives lp_solve_5.5_dev.(zip or tar.gz) and lp_solve_5.5_exe.(zip or tar.gz) to a standard library directory on the target platform. On Windows, the typical directory is \WINDOWS or \WINDOWS\SYSTEM32. On Linux, the typical directory is /usr/local/lib.
  - Unzip the Java wrapper distribution file to a new directory. On Windows, copy the wrapper stub library **lpsolve55j.dll** to the directory that already contains **lpsolve55.dll**.On Linux, copy the wrapper stub library **liblpsolve55j.so** to the directory that already contains **liblpsolve55.so**. Run **ldconfig** to include the library in the shared library cache.
  - You can look more details on the website (http://lpsolve.sourceforge.net/5.5/).
- The version of JRE requires 1.8 and above.
- The input documents must be encoded using UTF-8.

3. Usage

Open a terminal under the PKUSUMSUM directory and type in:
> java -jar PKUSUMSUM.jar <parameters>

Parameters:
There are several parameters required to be set when using the toolkit. Parameters in the "[]" are optional and they have default values.

========== **Required Parameters** ==========	
-T <type>	Specify which task to do. 1: single-document summarization; 2: multi-document summarization; 3: topic-based multi-document summarization.
-topic <topicFile>	Specify the path of the topic file only for the topic-based multi-document summarization task.
-input <inputPath>	Specify the path of the input document or document set. For single-document summarization, it specifies the path of the input document (including the document filename) to be summarized. For multi-document summarization or topic-based multi-document summarization, it specifies the directory of the input documents to be summarized.
-output <outputFile>	Specify the path of the output file containing the final summary.
-L <language>	Specify the language of the input document(s): 1 – Chinese, 2 – English, 3 - other Western languages.
-n <abNum>	Specify the expected number of words in the final summary.
-m <method>	Specify which method is used to solve the problem. For single-document summarization: 1 - Lead, 2 - Centroid, 3 - ILP, 4 - LexPageRank, 5 - TextRank, 6 - Submodular; For multi-document summarization: 0 - Coverage, 1 - Lead, 2 - Centroid, 3 - ILP, 4 - LexPageRank, 5 - TextRank, 6 - Submodular, 7 - ClusterCMRW; For topic-based multi-document summarization: 0 - Coverage, 1 - Lead, 2 - Centroid, 8 - ManifoldRank.
-stop <stopwordPath>	Specify whether to remove the stopwords. If you need to remove the stop words, you should provide the stopword list and specify the path of the stop word file. Note that we have prepared an English stopword list in the file "/lib/stopword_Eng", you can use it by input "y". If you don't need to remove the stop words, please input "n".

========== **Optional Parameters** ==========	
[-s <stemmerOrNot>]	Specify whether you want to conduct word stemming (Only for English language): 1 - stem, 2 - no stem; the default value is 1.
[-R <ReMethod>]	Specify which redundancy removal method is used for summary sentence selection. The ILP and Submodular methods don't need extra redundancy removal. The default value is 3 for ManifoldRank, and 1 for other methods which need redundancy removal. 1 – MMR-based method; 2 – Threshold-based method: if the maximum similarity between an unselected sentence and the already selected sentences is larger than a predefined threshold, this unselected sentence will be removed. 3 – Penalty imposing method: after a summary sentence is selected, the score of each

	unselected sentence will be penalized by subtracting the product of a predefined penalty ratio and the similarity between the unselected sentence and the summary sentence.
[-p <RePara>]	It is the internal parameter of the redundancy removal methods and has a default value of 0.7. For MMR and Penalty imposing method, it specifies the penalty ratio. For threshold-based method, it specifies the threshold value.
[-beta <beta>]	It is a scaling factor of sentence length when we choose sentences, and its range is [0, 1]. In several summarization methods, long sentences are likely to get higher scores than short sentences. Considering the length limit of the summary, we provide a scaling factor of sentence length to normalize the score of each sentence. $score'(s) = score(s)/length(s)^{beta}$, where *score(s)* is the initial score of sentence *s* calculated by the method you choose and $score'(s)$ is the normalized score which is used for sentence selection. Obviously, $score'(s) = score(s)$ when $\text{beta} = 0$. The default value is 0.1.
[-] LexPageRank-specific parameters	
[-link <linkThresh>]	It specifies the similarity threshold for linking two sentences. If the similarity of two sentences is larger than the threshold, then add an edge between the sentences. Its range is [0, 1] and the default value is 0.1.
[-] ClusterCMRW-specific parameters	
[-Alpha <AlphaC>]	It specifies the ratio for controlling the expected cluster number of the document set. Its range is [0, 1] and has a default value of 0.1.
[-Lamda <LambdaC>]	It specifies the combination weight for controlling the relative contributions from the source cluster and the destination cluster. Its range is [0, 1] and has a default value of 0.8.
[-] Submodular-specific parameters	
[-sub <op>]	It specifies the type of the submodular method, and the default value is 2. 1 – a method in Li's paper (Li at el, 2012); 2 - a modification method from Lin's paper (Lin and Bilmes, 2010);
[-A <AlphaS>]	It specifies the threshold coefficient. The range is [0, 1] and the default value is 0.5.
[-lam <LambdaS>]	It specifies the trade-off coefficient. The range is [0, 1] and the default value is 0.15 for multi-document summarization and 0.5 for single-document summarization.

License

Contact us

Welcome to contact us if you have any questions or suggestions while using PKUSUMSUM.
Contact person: Jianmin Zhang
Contact email: zhangjianmin2015@pku.edu.cn

Reference

[1]. Radev, Dragomir R., Hongyan Jing, and Malgorzata Budzikowska. 2000. *Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies.* Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization, Association for Computational Linguistics.

[2]. Mihalcea, Rada, and Paul Tarau. 2004. *TextRank: Bringing order into texts.* Association for Computational Linguistics.

[3]. Erkan, Günes, and Dragomir R. Radev. 2004. *LexPageRank: Prestige in Multi-Document Text Summarization. EMNLP*. Vol.4.

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

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

[6]. Lin, Hui, and Jeff Bilmes. 2010. *Multi-document summarization via budgeted maximization of submodular functions.* Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics.

[7]. Wan, Xiaojun, and Jianwu Yang. 2008. *Multi-document summarization using cluster-based link analysis.* Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval, ACM.

[8]. Wan, Xiaojun, Jianwu Yang, and Jianguo Xiao. 2007. *Manifold-Ranking Based Topic-Focused Multi-Document Summarization. IJCAI*. Vol. 7.