Extractive Multi-document Summarization System Based on LexRank

Harita Kannan
University of Washington
haritak5@uw.edu

Ben McCready
University of Washington
bmccr@uw.edu

Lonny Strunk
University of Washington
lstrunk@uw.edu

Xiaopei Wu
University of Washington
xw126@uw.edu

Abstract

For this project, we developed an extractive summarization system in a multi-document setting. We chose the LexRank algorithm as our content selection approach, and improved it with a stemmer. Additionally, we implemented an information ordering technique and summary post-processing. We evaluated our results on the Document Understanding Conference (DUC) datasets using the ROUGE metrics. Our results demonstrate that the enhanced content selection and added information ordering methods improve the performance of our system over the baseline of a naïve implementation of LexRank.

1 Introduction

This system converts sets of text-format newswire stories, already grouped by topic, into 100-word extractive text summaries. In Section 2, the general system architecture is described. In Section 3, we detail the changes made to the openly-available LexRank implementation (Shostenko, 2018) that the system was formed around in the venues of content selection, information ordering, and content realization are described. The quality of output as assessed by the ROUGE evaluation metric are presented in Section 4, followed by a discussion of the results and error analysis in Section 5.

2 System Overview

Our system, shown in Figure 1, has two major components; the Conductor module, which handles all file retrieval, preprocessing, information ordering, and post-processing and a separate LexRank module which does the work of content selection by generating summaries from sets of sentences.

The Conductor takes the topic list XML file as input: from there, it arranges the XML into a workable itinerary of Topic objects, each of which contains a set of documents associated with it. A Document object contains a list of sentences that are split using NLTK's sentence tokenizer[1].

All of the documents from all of the topics are then preprocessed by NLTK's Snowball Stemmer[2] and fed into the LexRank module to create the IDF numbers used to calculate the salience of given sentences.

After passing in a set of documents to be summarized by the LexRank module, a list of the sentences ordered by LexRank score are returned by the LexRank module, which are then passed to the information ordering function. We have implemented a greedy ordering technique. After the information ordering, the final summaries for the Topic objects are then written to output files.

3 Approach

This system integrates an openly available LexRank implementation (Shostenko, 2018) based on the concept introduced by Erkan and Radev (2004). The sentences that score highest according to LexRank are those which share more content with the other sentences among a given topic than do other sentences in that set, and this valuation most strongly informs which sentences summarize the topic best. Through iterative development cycles, additional modules were crafted to improve on the quality of these summaries.

3.1 Content Selection

After all texts from the corpus documents under a certain topic are collected into a textblock, the system processes the textblock by performing

[1] http:www.nltk.org/api/nltk.tokenize.html
[2] http:www.nltk.org/api/nltk.stem.html

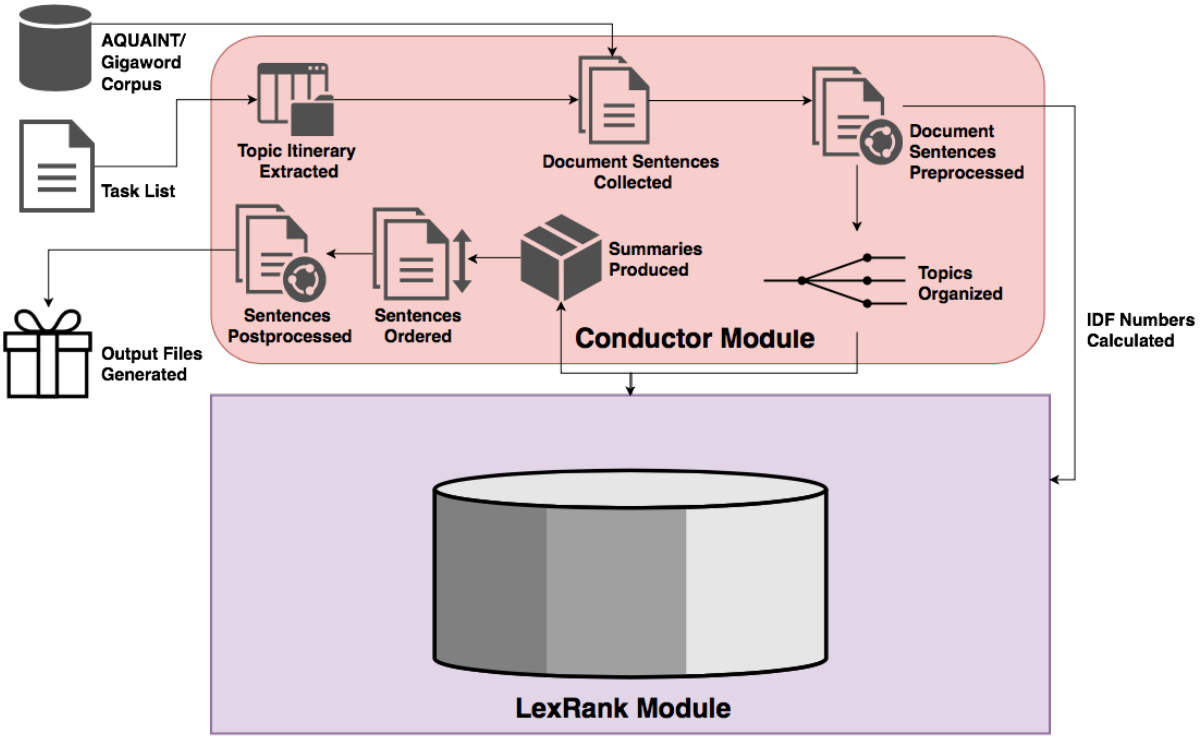

Figure 1: System Architecture

sentence tokenization using NLTK's sentence tokenizer[3], as well as preprocesses by stemming individual words in every sentence using NLTK's Snowball Stemmer[4]. The stemmed sentences from all of the documents are then passed into the LexRank module to calculate IDF scores.

When passing in a set of documents to be summarized by the LexRank module, we treat the document sets within a Topic object as one big document by appending their sentence lists together. The stemmed sentences from a single topic are then passed into the LexRank module again to calculate TF-IDF based cosine similarity scores, using a degree centrality threshold of 0.1. Various threshold values were tested but a threshold of 0.1 proved best in terms of ROUGE evaluation scores. Indices of k top-scoring stemmed sentences for each topic are returned, and the indices are then mapped onto the original sentences to extract the un-stemmed sentences for information ordering.

3.2 Information Ordering

For the core system, information was ordered according to the LexRank score; that is, of all sentences in a given group of documents, the sentences with the highest LexRank score were appended to our summary in order of score, until the summary cannot accept the next highest sentence without exceeding the maximum summary word length. After experimenting with a couple of more eloquent information ordering algorithms, we have added a 'greedy ordering' technique for information ordering.

3.2.1 Greedy Ordering

In replacement of the naïve ordering of the baseline system, our current system implements a simple greedy algorithm approach based on the ordering algorithm proposed by Nayeem and Chali (2017). The rationale is that a summary with good coherence will have similarity between all adjacent sentences. Furthermore, the presence of the same entities in adjacent sentences implies a coherent and ordered text. This approach takes as input the k top ranked sentences from the LexRank output. These sentences are then incrementally added to a document D which is initially empty. The sentence to be placed is considered for all the positions in the document and the coherence of the document as a whole is calculated for each placement for each sentence. The document ordering with maximum coherence is returned. Coherence of document D is calculated by computing sum of

[3]See footnote 1.
[4]See footnote 2.

pairwise similarity of sentences in D. The similarity metric is based on TF-IDF modified cosine.

3.3 Content Realization

Being that this system produces an extractive summary, all content included in the system output are sentences chosen verbatim from documents within the document set of each topic for which the summary is being produced. Content realization in the form of sentence compression was implemented but was not incorporated in the final system as the readability of the resulting summaries declined when used. We integrated an available Multi-Sentence Compression implementation (Boudin, 2014) based on the algorithm introduced by Filipova (2010). This module, which was run after information ordering and other preprocessing steps, took pairwise sentences from a temporary summary as input and returned a compressed sentence. Our system did not perform well with this method mainly due to two reasons. Firstly, the lack of sentence clustering in the content selection process meant input sentences were not guaranteed to be suitable for compression. A better option would be to implement sentence compression paired with sentence clustering during the content selection stage. Secondly, the algorithm focuses on extracting only keywords and would compress key points into ungrammatical sentence fragments.

4 Results

The test data for our implementation was taken from the Document Understanding Conferences (DUC) 2010 AQUAINT/Gigaword corpus (Graff, 2002; Parker et al., 2011) and used the ROUGE evaluation metric to compare our system performance. The baseline system consists of just the LexRank module without the stemming and using the LexRank score for information ordering. We then ran the system with greedy ordering and stemming. The addition of stemming and ordering vastly improved ROUGE scores and readability of the summaries compared to the baseline.

Table 1 presents ROUGE scores for the previous development cycle's system, run over the development dataset, before sentence compression was implemented. Table 2 reflects the ROUGE scores of this same dataset. While ROUGE scores are similar between table 1 and 2, the quality of output improved significantly in quality when reviewed by fluent human judges. Table 3 presents ROGUE scores for the final system run over the test dataset.

Table 1: D3 ROUGE Scores

	PRECISION	RECALL	F1
ROUGE-1	0.30199	0.23441	0.26228
ROUGE-2	0.07412	0.05774	0.06445
ROUGE-4	0.00600	0.00489	0.00534

Table 2: Development Dataset ROUGE Scores

	PRECISION	RECALL	F1
ROUGE-1	0.30241	0.23798	0.26502
ROUGE-2	0.07397	0.05785	0.06457
ROUGE-4	0.00756	0.00589	0.00657

Table 3: Test Dataset ROUGE Scores

	PRECISION	RECALL	F1
ROUGE-1	0.32381	0.26707	0.29103
ROUGE-2	0.09143	0.07698	0.08316
ROUGE-4	0.01480	0.01229	0.01336

5 Discussion

The hypothesized improvements to the bare LexRank algorithm resulted in higher-quality topic summaries, both qualitatively as reviewed by the authors and quantitatively, according to the resultant ROUGE scores.

5.1 Improvements

Output summary of a specific topic from the development dataset is exemplified in Example 2. Compared to the output of our system without compression on the same topic (Example 1), topic relevancy as well as readability has improved. Our system's modified content selection and added post-processing have filtered out most of the sentence fragments and all news headers which were present in previous development cycles; all reduplicated sentences have also been removed from the outputs in the post-processing step. Overall cohesion has also improved with the help of information ordering.

Example 1: D3 Sample Summary Output from Development Dataset

The province has limited the number of trees to be chopped down in the forest area in northwest Yunnan and has stopped building sugar factories in the Xishuangbanna region to preserve the only tropical rain forest in the country located there.
HAIKOU, June 22 (Xinhua) – A tropical rain forest project is to start soon in south China's Hainan province.
KUNMING, June 23 (Xinhua) – Xishuangbanna, one of China's largest tropical rain forest reserves, will almost double its area to bring more wild plants and animals under protection.
Gibbons are regarded as an animal indicator in the tropical rain forest.

Example 2: Sample Summary Output from Development Dataset

Xishuangbanna, one of China's largest tropical rain forest reserves, will almost double its area to bring more wild plants and animals under protection.
A tropical rain forest project is to start soon in south China's Hainan province.
The province has limited the number of trees to be chopped down in the forest area in northwest Yunnan and has stopped building sugar factories in the Xishuangbanna region to preserve the only tropical rain forest in the country located there.
Gibbons are regarded as an animal indicator in the tropical rain forest.

Example 3: Sample Summary Output from Test Dataset

The owner of a Saudi oil supertanker hijacked by Somali pirates over the weekend said the 25 crew members are safe and the ship is fully loaded with crude – a cargo worth about $100 million at current prices.
The owner of a Saudi oil supertanker hijacked by Somali pirates over the weekend says the 25 crew are safe and that the ship is fully loaded with crude.
Christensen said the pirates hijacked the vessel on Saturday.
The Sirius Star is the largest ship ever taken by Somali pirates, though large chemical tankers and freighters have also been hijacked.

5.2 Error Analysis

Output summary of a specific topic from the test dataset is exemplified in Example 3. While Example 3 is a reasonably informative and readable summary, several issues still remain. The most prolific and prominent issue seen in system output is redundancy; this is due to identical or nearly identical sentences across different documents under the same topic in the corpus, and the fact that our greedy information ordering approach often lines them up together because of its coherence-based metrics. Although the post-processing step removes sentences that are exact duplicates of existing ones, it does not prevent sentences that are highly similar (such as the first and second sentences in Example 3) from appearing together. The issue could potentially be resolved by implementing a more extensive post-processing step that blocks sentences with a high word co-appearance ratio, or by adding sentence-clustering in the content selection process by grouping similar sentences together and only selecting one sentence from each cluster.

Another issue is "out-of-order" segments, in which definite subjects (e.g. "the oysters") appear in the summaries prior to their introduction, which appears either later on or not at all. Other summaries featured out of place transition words (i.e. beginning a summary with "however"). These are issues to be dealt with via content realization strategies in future iterations of this system.

Additionally, our outputs do not always fully make use of the 100-word constraint. Our system

stops outputting the subsequent sentence to a summary if its length makes the the current summary exceed 100 words. Consequently, this could result in small or even empty summaries if the top selected sentences are very long. This could be resolved by modifying the post-processing step to iterate down the top sentence list until the 100-word limit is filled.

Finally, some of the current summary outputs still contain errors from the baseline outputs, such as erroneously segmented sentences caused by abbreviations like "i.e." and "Gov.", divided on account of the period followed by a space. This could be rectified with a list of common English abbreviations or the implementation of a better sentence tokenizer.

5.3 Future Work

In the future we plan to modify the following aspects of our system: (1) Augmenting the current content selection component with topic-oriented mechanisms to further focus on a given topic. This would choose sentences that focus on the main topic instead of minor topics. (2) Adding sentence clustering in the system's content selection process to group together similar sentences. Selecting one sentence from each cluster could reduce redundancy, while selecting sentences from different clusters could increase the summary's coverage of the topic and its informativeness. (3) Further improve the post-processing step by utilizing sentences lower in the sentence list to maximize the word count summary given the 100-word constraint.

References

Florian Boudin. 2014. takahe. `https://github.com/boudinfl/takahe`.

Günes Erkan and Dragomir R. Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22(1):457–479, December.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 322–330.

David Graff. 2002. The aquaint corpus of english news text ldc2002t31. Web Download.

Mir Tafseer Nayeem and Yllias Chali. 2017. Extract with order for coherent multi-document summarization. In *Proceedings of TextGraphs-11: the Workshop on Graph-based Methods for Natural Language Processing*, pages 51–56, Vancouver, Canada, August. Association for Computational Linguistics.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition ldc2011t07.

Luka Shostenko. 2018. lexrank. `https://github.com/wikibusiness/lexrank`.