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 aim to develop 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 two information ordering techniques. 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

The objective of this system is to convert 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, justification for 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, and information ordering, 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 two different information ordering techniques: chronological ordering and greedy ordering. 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 inspiration for choosing the LexRank algorithm to form the core of the system was the conceptual analogy can be clearly drawn between the summarization task at hand and the way that LexRank rewards sentences that relate to the rest of the sentences within a document set; 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.

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

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

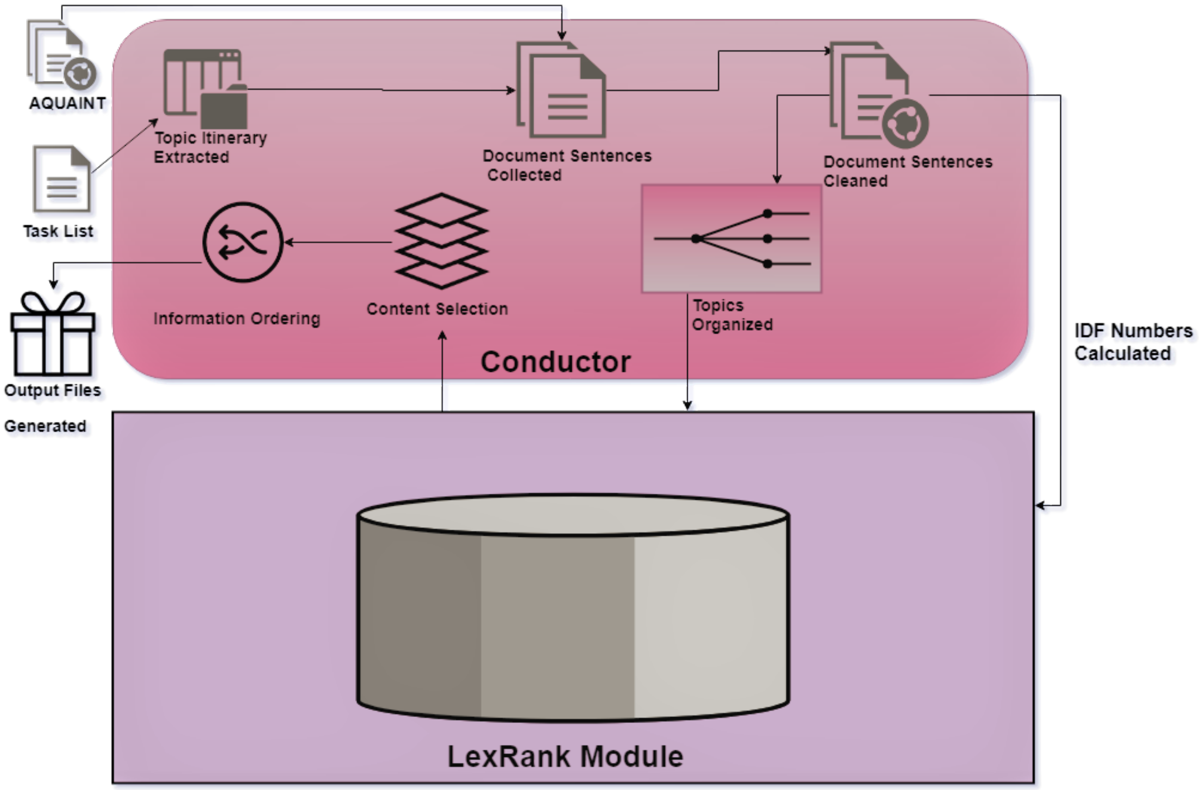

Figure 1: System Architecture

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 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 baseline 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 our summary cannot accept the next highest sentence without exceeding the required word length. We have implemented two other techniques for information ordering: chronological ordering and greedy ordering.

3.2.1 Chronological Ordering

A rule-based chronological information ordering function was added to the baseline system, which used extracted publication date information from the documents to order the top k sentences selected by LexRank from earliest to latest. For technical reasons, the system's chronological ordering variant did not make use of the stemming feature.

3.2.2 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

[3] See footnote 1.

[4] See footnote 2.

(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 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.

4 Results

The test data for our implementation was taken from the Document Understanding Conferences (DUC) 2010 AQUAINT/AQUAINT-2 corpus (Graff, 2002; Vorhees and Graff, 2008) 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 chronological ordering (no stemming) and finally ran the system again with greedy ordering and stemming. Tables 1-3 present ROUGE scores for the three systems. The baseline uses $k = 10$ for the top k sentences returned by the LexRank module. The two non-baseline systems use $k = 5$ as this resulted in improved scores.

Table 1: Baseline ROUGE Scores

	PRECISION	RECALL	F1
ROUGE-1	0.17634	0.22780	0.19742
ROUGE-2	0.03692	0.04760	0.04127
ROUGE-4	0.00303	0.00421	0.00346

Table 2: Chronological Ordering ROUGE Scores

	PRECISION	RECALL	F1
ROUGE-1	0.29408	0.23697	0.26133
ROUGE-2	0.08084	0.06513	0.07185
ROUGE-4	0.00932	0.00775	0.00843

Table 3: Greedy Ordering 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

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 Error Analysis

Output summary of a specific topic from the system described is exemplified in Example 2. Compared to the output of our baseline system on the same topic (Example 1), topic relevancy as well as readability has improved. However, several issues still remain. The most prolific issue seen in system output is redundancy; reduplicated sentences exist in some of our output summaries (see the second and third sentences in Example 2). This is due to identical sentences across different documents under the same topic in the corpus, and our greedy information ordering approach often lines them up together based on its coherence-based metrics. This could potentially be resolved by adding sentence-clustering in the content selection process by grouping similar sentences together, and only selecting one sentence from each cluster.

Example 1: Baseline Sample Summary Output

> By TOM JONES.
> Felling said.
> Among those receiving subpoenas is Joe Lebrun, who was involved in the Texas lawsuit accusing Citizens' former chief operating officer Paul Hulsebush with a bribery scheme.
> BC-STEIN (St. Petersburg) – A Q&A with the visiting Ben Stein, economist/Nixon White House protege/actor/author/political commentator.
> "Is you the first girl she messed with?"
> "If you had to sit back and just from a societal point of view, (ask) what impact does the media have by putting such focus on this case when a young boy is going to be affected, that's the balancing act."

Example 2: Sample Summary Output

> Lafave's lawyer, the veteran and able John Fitzgibbons of Tampa, objected because the state wished her to serve what he considered an unacceptable amount of prison time.
> After Tuesday's hearing, Lafave's attorney, John Fitzgibbons, said the plea was "a fair resolution of this case."
> After Tuesday's hearing, Lafave's attorney, John Fitzgibbons, said the plea was "a fair resolution of this case."
> Fitzgibbons said in July that plea negotiations had broken off because prosecutors insisted on prison time, which he said would be too dangerous for someone as attractive as Lafave.

While both ordering algorithms implemented improved the quality of summaries over those generated by the bare LexRank package, both variants still generated "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.

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. News headers such as "BC-STEIN (St. Petersburg)" in Example 1 are still present in some summaries as well.

5.2 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.(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 informativity. (3) Conducting more experiments and more in-depth error analysis to determine the proper information ordering approach. (4) Implementing post-processing techniques to improve the quality of output summaries, such as removing news headers. (5) Implementing sentence compression or sentence reformulation to enhance content realization and further improve the readability of output summaries.

References

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.

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.

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

Ellen Vorhees and David Graff. 2008. Aquaint-2 information-retrieval text research collection ldc2008t25. Web Download.