573 Deliverable 2: A Baseline for Automatic Extractive Summarization

Ayushi Aggarwal
University of Washington
ayushiag@uw.edu

Elijah Rippeth
University of Washington
rippeth@uw.edu

Seraphina Goldfarb-Tarrant
University of Washington
serif@uw.edu

Abstract

We developed an end-to-end baseline for an extractive text summarization pipeline for the Text Analysis Conference shared task. Three approaches were explored, namely, first-sentence, random-sentence, and LexRank - as baselines for future work. We found that the first-sentence approach outperformed random-sentence by as much as 50% in F1 in ROUGE scores and LexRank by as much as 35.4% in F1 in ROUGE scores, though LexRank generally was judged qualitatively superior. Approaches to supplement LexRank with better summarization capabilities are also discussed along with possible improvements to the pipeline in general.

1 Introduction

The Text Analysis Conference(TAC)[1] summarization task provides a large collection of newswire articles to be compressed into short(100 words) summaries by topic. As per the task, the summaries are evaluated for content and responsiveness. With this goal in mind, we approach automatic summarization of the given dataset. The two main approaches by which text summarization is realized are abstractive, which focuses on capturing concepts and generating potentially novel utterances(paraphrases), and extractive, which filters out the least salient sentences from a corpus, leaving only the most relevant information.

Given the time-frame for the development of this system, we attempt to develop an extractive summarization system. In this paper, we explore the various baselines for content selection and ordering, develop a working end-to-end baseline and evaluate the summary outputs for quality by looking at coherence, informativeness and anaphora resolution. We also present the results of ROUGE score analysis and hyper-parameter tuning as detailed in the Results and Discussion sections.

1.1 System Architecture

The main modules of the architecture are enumerated below with details of third-party packages and APIs they utilize. Details of the functionality are provided in the Approach section.

1. **Corpus Reader:** This module is tasked with reading in the dataset as provided for the TAC shared task. It utilizes lxml[2] and BeautifulSoup[3] for XML parsing and passes the resultant to the preprocessor module. PyYAML was used for custom end-to-end system configuration.

2. **Preprocessor:** The preprocessor resides within the corpus reader as as a method and stores all documents for a topic as story data structures that further have the heading, sentences and spans attributes. For each topic, sentences are tokenized using the NLTK[4] Punkt Sentence Tokenizer and TreeBank Word Tokenizer.

3. **Configurable summarizer:** This is the main module in our pipeline that creates a single list of all sentences for a topic. Stemming is then performed using the NLTK Porter Stemmer*http://www.nltk.org/howto/stem.html*. It utilizes a customized fork of the sumy[5] implementation for LexRank technique for computing sentence saliency scores. This is further detailed upon in the section that follows.

[1] https://tac.nist.gov/tracks/index.html

[2] http://lxml.de/

[3] https://www.crummy.com/software/BeautifulSoup/

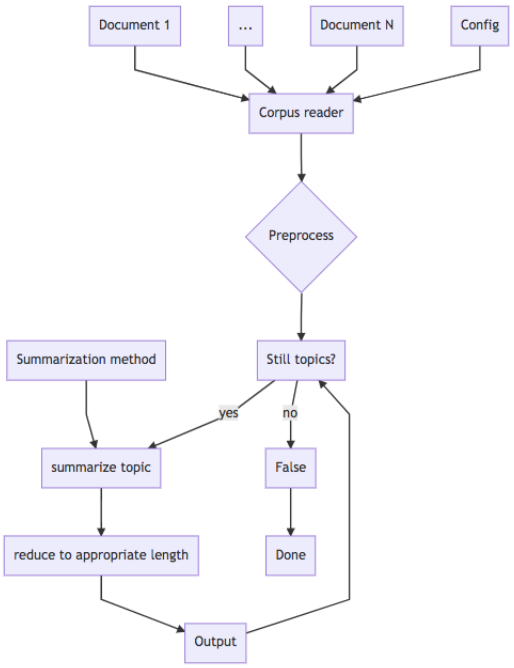
[4] https://www.nltk.org/

[5] https://github.com/miso-belica/sumy/tree/dev/sumy/summarizers

Numpy was utilized for linear alegbra calculations as a part of LexRank.

4. **Post-processing scheme:** This scheme generates the final summary as expected by the TAC task definition.

A wire diagram of the system is shown below:



2 Approach

This section details the functionality of the core components of the summarization pipeline.

2.1 Dataset and Preprocessing

In the interest of making the system as portable as possible and for the ease of parallel project group work, we operated on configuration files from the onset. The TAC dataset is categorized by topic, with each topic consisting of a set of documents. Since the task is topic-orientated and the ordering of documents does not matter, all documents are combined into one giant document for each topic. The **corpus reader** stores the pointers to the relevant documents in the corpus grouped by topic set. This allows us to avoid the potentially expensive task of storing entire documents and operate solely on a subset of the topics covered by the dataset.

The **corpus reader** proved to be the most tedious and time-consuming module to build in the baseline pipeline. Inconsistencies with formatting of documents in different datasets required separate ingest mechanisms, which proved to be fairly error-prone and thereby, a time drain.

2.2 Content Selection

The system baseline takes the sentences for each topic story as the input to the **summarize** module. We implemented a quick-and-dirty baseline that chooses the sentences that occur first in the set of sentences per topic and generates a summary. The final baseline of the system implements a graph based sentence saliency score computation method called LexRank(Ergan and Radev, 2004). All sentences in a topic act as the nodes of the graph. Weights for the edges are calculated by considering sentences as bag-of-word and computing the tf-idf cosine similarity between sentences and centrality score for each sentence node v computed as the fraction of the sum of the centrality scores of all the neighboring nodes to the degree of the node. Further, a matrix of the lexRank scores is calculated using the centrality score for each sentence. We chose LexRank for its easily configurable nature and ease of availability.

2.3 Information Ordering

Once lexRank scores are obtained for all sentences for a topic, sentences are arranged in descending order of scores. Given the nature of the dataset, we analyzed the possibility of including publishing date for stories. However, since we found the date information to be sparse, temporal information was not incorporated into the information ordering for the final summary.

2.4 Post-processing

The ordered output obtained from LexRank is further processed to adhere to the final summary word limit of 100. We include the highest scoring first n sentences that fall within the word-limit. The user can modify the summary word-limit for each topic by resetting the hyper-parameter *word_num*.

3 Evaluation Results

Tables 1, 2 and 3 depict the results for experiments run on the DevTest data given for TAC.

Overall, it was surprising to see such low ROUGE-1 scores for all of the approaches. Our ROUGE scores were well below those reported in the LexRank paper(Ergan and Radev, 2004). In contrast, the first-sentences approach outperformed the other two baselines in every category apart from ROUGE-1 Precision, wherein LexRank narrowly beat the first-sentences approach.
First-sentences and LexRank have very high

ROUGE scores with respect to each other, even though they are qualitatively quiet different. This is further discussed in the following section. Table 4 shows the impressive level of unigram and even 4-gram overlap between the results of the two systems. For the given corpus, selecting the most connected sentences often ends up selecting the first one, though no position information is taken into account as documents are considered to be bags of sentences.

Effect of tuning the threshold hyperparameter for LexRank edge weights was also studied. As shown in Table 5, the best ROUGE-1 scores are obtained for threshold value of 0.1.

Type	P	R	F1
ROUGE-1	20.39	16.77	18.28
ROUGE-2	4.37	3.69	3.98
ROUGE-3	1.35	1.18	1.25
ROUGE-4	0.59	0.52	0.55

Table 1: ROUGE scores for random-sentence baseline run on DevTest data

Type	P	R	F1
ROUGE-1	26.91	23.08	24.68
ROUGE-2	8.13	6.99	7.46
ROUGE-3	2.65	2.32	2.46
ROUGE-4	1.17	1.04	1.10

Table 2: ROUGE scores for first-sentence baseline run on DevTest data

Type	P	R	F1
ROUGE-1	28.95	21.98	24.75
ROUGE-2	7.70	5.82	6.57
ROUGE-3	2.49	1.91	2.14
ROUGE-4	0.81	0.64	0.71

Table 3: ROUGE scores for LexRank baseline run on DevTest data

Type	P	R	F1
ROUGE-1	52.49	51.88	51.71
ROUGE-2	41.13	40.16	40.30
ROUGE-3	36.69	35.58	35.82
ROUGE-4	33.87	32.77	33.01

Table 4: LexRank vs. First

Threshold	P	R	F1
0.1	29.239	21.805	24.693
0.2	28.185	22.115	24.55
0.4	22.616	18.057	19.99
0.5	21.673	16.879	18.877
0.9	21.286	16.572	18.555

Table 5: ROUGE-1 scores for different thresholds for LexRank

4 Discussion

As expected for inverted pyramid structure of Newswire corpus content, the baseline of selecting first-sentences is hard to beat. It was in fact, *harder* to beat than we anticipated, as LexRank does not outperform first-sentences in any area save ROUGE-1 Precision, in which the difference is not large enough to be statistically significant.
However, the results were qualitatively different. Many first sentences are a type of short abstract summarizing all topics that follow, but some are a specific appeals to emotion or descriptions of an incident that are meant to introduce the feel of the narrative. For example, in summaries about the repeal of a pharmaceutical drug, some first sentences look like, *'When Emily Martin was hospitalized for emergency gallbladder surgery last summer, her doctors found that she had also had acid reflux, causing erosion of her esophagus.'* Where *Emily Martin* is not actually important and does not appear later, or in the model summaries; she is there to provide a human element to a broader story. In these cases, another standard narrative convention works *against* the first-sentences heuristic, just as the pyramidal structure works towards it.
LexRank avoids this problem by valuing connectedness; these sentences are very detailed, and positionally important, but not very connected.
The downside of the LexRank structure is the product of its components. Cosine similarity between sentences will bias towards longer ones, and tf-idf will overweight rare words. Our experiments display this tendency. We extracted statistics on sentence length and on lexical diversity (defined as the ratio of unique to total words, per document) and performed the Mann-Whitney U test on the distributions over the model summaries and the LexRank summaries. We pre-selected $p = 0.01$ and found that the difference between both distributions easily passed the significance test. The

means of the results are in Table 4. Since the model summaries are drawn from seven separate human summarizers, and are still so consistently shorter in sentence lengths and poorer in lexical diversity, it may be rewarding to look into ways to better approximate the style of human summarizers. That said, such a change would come with costs - for instance, shorter sentences will require more anaphora and co-reference resolution.

A weakness of all of the baselines is their lack of any kind of discourse coherence, of anaphora resolution, or of penalties for redundancy (literal string repetition, or concept repetition via paraphrase). As noted, we did not have an ordering component to the system save "best scores" from LexRank. However, ordering would have little effect on ROUGE scores, and had little effect on qualitative evaluation as well. When discourse coherence was lacking, it was not just do to ordering, but to necessary information being missing. Therefore, discourse coherence and anaphora resolution seem more promising areas for future improvement. Methods of event extraction to ensure only one sentence per event, or of distributional semantics to address synonymy, will certainly improve performance qualitatively by reducing redundancy, though it is uncertain what the result will be on ROUGE Recall scores. One simple iterative improvement would be to experiment with addressing redundancy and the bias towards rarity by switching from tf-idf vectors to embedded sentence (or sub-sentence) representations trained on a larger corpus.

All baselines also share the weaknesses of their pipeline components, particularly of sentence tokenization. Qualitative analysis revealed some irregularities in sentence tokenization, which may be possible to correct via training a domain specific Punkt Tokenizer. As this is an unsupervised algorithm, this would not be prohibitively difficult or expensive.

Despite LexRank sentence choices being qualitatively salient, much more so than first-sentences, they did not always display good recall of events/concepts. Of X concepts dispersed amongst the original ten documents (ten per topic), sometimes LexRank will select only a small subset of X. For example, in this summarization of a murder, all the details of the actual murder are missing (See Appendix). This is a source of lack of coherence, as there are missing causal links in the chain, and also renders summarizations useless in practice. Summarization as a lossy compression system for transmitting large amounts of information in bitesize pieces should have strong objectives for concept and/or event recall. One of the methods of redundancy, event extraction, could also be used to penalize poor event recall. This will be a necessary metric to incorporate into future efforts, that is also algorithm independent.

Type	**Avg Words**	**LDR**
Model	20.19	0.69
LexRank	28.25	0.73

Table 6: Distributional Statistics. **Avg Words** are per sentence, **LDR** is Lexical Diversity Ratio.

5 Conclusion

In this work, we hypothesized that while the first-sentence summary would be a competitive baseline due to the inverse pyramid structure of the data, LexRank would outperform the naive heuristic. Indeed, this was shown to not be the case for our data. The first-sentence baseline outperformed all other baselines in nearly all ROUGE metrics.

Extensive experimentation with baselines indicates that they could be improved upon by increasing discourse coherence by penalizing redundancies summary sentences which would minimize paraphrases of the same salient topics. Additionally, topic modeling, improving event coverage, and anaphora resolution in output could generate more content rich and representative summaries.

Acknowledgments

We would like to specially thank Noam Chomsky, without whom we would likely have no real basis for language. We are eternally indebted to you, Noam.

References

Allahyari, M., Pouriyeh, S., Assefi, M., Safaei, S., Trippe, E. D., Gutierrez, J. B., and Kochut, K. 2017. *Text summarization techniques: A brief survey.* arXiv preprint arXiv:1707.02268.

Brin, S., Page, L., Motwani, R., and Winograd, T. 1998. *The PageRank citation ranking: Bringing order to the web.* Stanford InfoLab.

Erkan, G., and Radev, D. R. 2004. *Lexrank: Graph-based lexical centrality as salience in text summarization.* Journal of Artificial Intelligence Research, 22, 457-479.

Nenkova, A., and McKeown, K. 2004. *A survey of text summarization techniques.* In Mining text data (pp. 43-76). Springer, Boston, MA.

6 Appendix

6.1 LexRank Examples

Note that all of the events of the actual murder are missing.

```
critics of prosecutors here
said that they had little
experience investigating and
trying homicides in boulder, a
college city that has about one
murder a year.
at the same time, beckner said
he was ``excited ''about new
evidence.
so when the ramseys had three
days of interviews last week
with investigators from the
district attorney's office,
gordon wondered, ``why the wait?
the ramsey interviews arrive as
boulder, colo., district attorney
alex hunter decides whether to
take the case to a grand jury.
``they may be posturing; they
may be trying to help, ''mueller
said.
```