Automatic Extractive Summarization for NewsWire Data

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 extractive text summarization system for the Text Analysis Conference[1] shared task. We utilized LexRank (Erkan and Radev, 2004) to compute sentence salience scores, supplemented with various embedding techniques, including TF-IDF, GloVe vectors, and Doc2Vec vectors. Additionally, a chronological content ordering expert adds to the general coherence of candidate sentences. We found that the TF-IDF embeddings outperformed the next most competitive embedding measure by as much as 37.6% in F1 in ROUGE scores, though other embeddings were judged to be qualitatively superior.

1 Introduction

The Text Analysis Conference (TAC)[1] summarization task provides a large collection of newswire articles to be compressed into short (no more than 100 words) summaries by topic. Each topic is given, and contains ten documents from disparate news sources, all in English. 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.

In this paper, we develop an extractive summarization system. 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.

1.1 System Architecture

The main modules of the system are illustrated in figure 1. This section highlights the third-party packages and APIs utilized by each module. Details of the functionality are provided in Section 2.

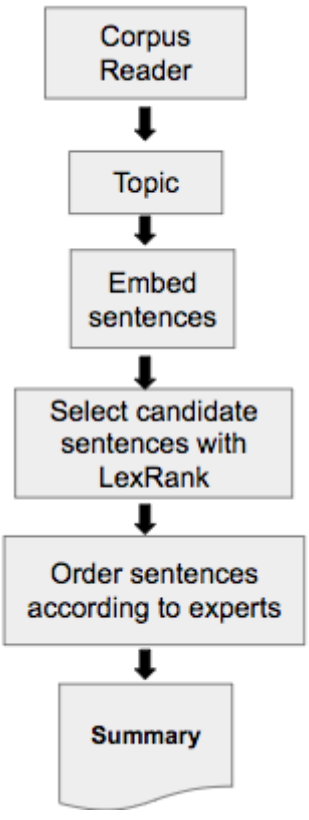

Figure 1: The system summarizing a single topic with an arbitrary sentence representation.

1. **Corpus Reader:** This module is tasked with reading in the dataset as provided for the TAC

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

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. A preprocessor resides within the corpus reader as as a method and stores all documents for a topic as story data structures that further have headings and Sentence objects. Sentence objects store metadata about the sentence (order in original document, date of original document) and configurable, arbitrary sentence embedding objects. For each topic, we experimented with tokenization via using both the NLTK[4] Punkt Sentence Tokenizer (Kiss and Strunk, 2004) and TreeBank Word Tokenizer and spaCy's[5] default and dependency parse tokenizers. All tokenizers available report high reliability across Precision, Recall, and F1, but given that LexRank operates with a sentence as a basic unit, and that it operates on relatively small numbers of sentences, even small errors in tokenization have the potential to affect performance significantly, which we found to be the case empirically in our initial observations of the system.

2. **Configurable summarizer:** This is the main module in our pipeline, which consumes a topic and outputs a ranked list of candidate sentences. It does further pre-processing if necessary for a given sentence embedding strategy. When stemming is performed we use the NLTK Porter Stemmer[6] (Porter, 1980). The summarizer utilizes a customized fork of the sumy[7] implementation for LexRank technique for computing sentence saliency scores. This is further detailed upon in the section that follows. Numpy was utilized for linear algebra calculations as a part of LexRank.

3. **Information orderer:** The information ordering module orders a set of candidate sentences which are the output of the summarizer according to various metadata. This is all original code which relies on extension of an expert interface for various ordering conditions. This will be discussed in detail in a later section.

[2] http://lxml.de/

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

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

[5] https://spacy.io/usage/linguistic-features

[6] http://www.nltk.org/howto/stem.html

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

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 flexible as possible for different experimental setups, 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 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 pipeline. Inconsistencies with formatting of documents in different datasets required tailored document ingest mechanisms, which proved to be fairly error-prone.

2.2 Content Selection

The system takes the sentences for each topic as the input to the **summarize** module. Because of the structured nature of the domain newswire articles, where the beginnings of documents often are natural summaries, we implemented a quick-and-dirty baseline that chooses the first sentence from each story in the topic and generates a summary while the summary is under the 100 word limit. We used this as our base metric which would be improved. We also implemented a baseline that selects sentences randomly from each document, to establish a performance lower bound. The final system implements a graph based sentence saliency score computation method called LexRank (Erkan and Radev, 2004), which is a modification of the PageRank algorithm (Brin and Page, 1998). All sentences in a topic act as the nodes of the graph. Weights for the edges

are calculated by considering sentences as bag-of-words 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. A matrix of the LexRank scores is calculated using the centrality score for each sentence, from which we can select the top *n*.

Because LexRank qualitatively tends to favor long sentences, we experimented with a sentence normalization penalty in the LexRank selection routine wherein the similarity will be normalized by the length of the shorter sentence when comparing two nodes in the graph walk.

One of the initial reasons we chose LexRank was its ease of adaptation to different experiments. Abstractly, LexRank operates based on sentence similarity, ergo the LexRank framework can be used on any set of sentences, given some way of calculating a similarity score between them. We experimented with substituting the TF-IDF representations of sentences with other options. We tried averaged GloVe embeddings (Pennington, 2014), provided by spaCy[8], which are trained on the Common Crawl Corpus.[9]. We also tried Doc2Vec (Le and Mikolov, 2014), trained via Gensim[10] on the English Gigaword Corpus[11]. We built the **summarize** module to be easily extensible to any new models of representing sentences, should it make sense to try more in future, though TF-IDF so strongly outperformed both GloVe and Doc2Vec that we chose not to pursue additional sentence representation variations at this time. We release our code to the public[12] to further research in this area.

2.3 Information Ordering

Information adding was added as a post-processing measure in hopes to improve coherence between sentences from disparate stories. To this end, we leverage a weighted linear sum of expert systems, first proposed in (Bollegala et al., 2012), to determine whether sentence u or v should be preferred given a partial summary Q:

$$p(u, v, Q) = \sum_{e \in E} w_e \cdot q(e, u, v, Q)$$

where E is the set of experts, w_e is the contribution a given expert makes to a preference, and q is a function with a domain [0, 1] that defines the amount of preference to be given to sentence e.

By defining ϵ as the threshold above which to prefer u, we can order all sentences in a document by iterating over the unique pairs (u, v), of combinations of all sentences and voting for u if $p(u, v, Q) > \epsilon$ and v otherwise. The order is determined by a reverse sort of votes.

To study how ordering helps improve the system, we defined a single information ordering expert based on (Bollegala et al., 2012) for chronological ordering. This method relies on metadata from the articles from which the sentences are extracted; specifically, it relies on document id ($D(x)$ is defined to be a function returning the document id of sentence x), publication date ($T(x)$ is defined to be a function returning the date the document containing sentence x was published), and the position of the sentence in a document ($N(x)$ is defined to be a function returning the index of a sentence in a document).

Ordering of the chronological expert is as follows:

$$q(c, u, v, \_) = \begin{cases} 1 & T(u) < T(v) \\ 1 & [D(u) = D(v)] \wedge [N(u) < N(v)] \\ 0.5 & [T(u) = T(v)] \wedge [D(u) \neq D(v)] \\ 0 & otherwise \end{cases}$$

In other words, if a sentence u was published before sentence v, u should be preferred earlier in a summary. Similarly, if two sentences are from the same document, the earlier sentence in the document should be preferred. If two unique documents were published on the same date, no preference should be given. Otherwise, prefer v.

With this single, simple heuristic, we noticed improved qualitative coherence in our summaries. We hypothesize that other experts may improve coherence additionally.

2.4 Post-processing

The ordered output obtained from LexRank is further processed via truncation at the sentence level to adhere to the final summary word limit of 100. We include the highest scoring first *n* sentences

[8] https://spacy.io/models/en#en_vectors_web_lg

[9] http://commoncrawl.org/the-data/

[10] https://radimrehurek.com/gensim/models/doc2vec.html

[11] https://catalog.ldc.upenn.edu/ldc2011t07

[12] https://github.com/erip/ling573-text-summarization

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

3.1 Initial Round of Experiments for Baseline

Tables 1, 2 and 3 depict the results for experiments run on the DevTest data given for TAC for the random, first-sentences, and LexRank systems. Henceforth, unless otherwise stated, LexRank scores refer to LexRank with its best-performing configuration: TF-IDF, with a threshold of 0.1. ROUGE scores are reported as the mean of multiple runs, as neither resolution of ties for LexRank nor post-processing truncation is deterministic, so different runs of the system will display small, non-significant differences in results.

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 (Erkan 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.
Table 4 shows how first-sentences and LexRank have very high ROUGE scores with respect to each other[13], even though they are qualitatively quite different (qualitative differences are discussed further in the following section). The results show the impressive level of unigram and even 4-gram overlap between the outputs of the two systems. For the given corpus, selecting the most connected sentences often ends up selecting the first one, though no sentence positional information is taken into account.

The 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.

3.2 Sentence Normalization Experiments

We observed that LexRank tended to prefer long sentences as compared to the model summary sentence lengths (see further discussion in 4.1). When our objective is to maximize saliency while minimizing the number of words, we hypothesized

[13]ROUGE scores measure overlap between a text and a reference (or multiple reference) texts. By setting first-sentences as the reference text (rather than model summaries) ROUGE shows the n-gram overlap between the output of the two systems.

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. (mean of 10 runs).

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: ROUGE scores between LexRank and first-sentences, on DevTest data

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 baseline

Type	P	R	F1
ROUGE-1	18.71	14.38	16.18
ROUGE-2	3.26	2.55	2.85
ROUGE-3	0.92	0.74	0.82
ROUGE-4	0.32	0.26	0.29

Table 6: ROUGE scores for sentence-normalized LexRank baseline run on DevTest data

that normalizing sentences might serve as a regularizer. When computing the similarity between two nodes in the LexRank graph, we experimented with normalization by the length of the shorter of the sentences. The results of normalization on DevTest can be seen in Table 6. Because results were strictly worse than the LexRank TF-IDF baseline, we decided not to include the normalization in the final system.

3.3 Sentence Embedding Experiments

We anticipated some weaknesses of TF-IDF - namely, that it should overvalue rare words, and that it captures only relatively local information as it treats sentences as unit vectors weighted by their TF-IDF scores within a sentence and document cluster. To ameliorate these, we extended our system to utilize GloVe and Doc2Vec embeddings. The results are summarized in Table 7.

We chose to use spaCy's GloVe embeddings because of their ability to capture global collocations, as the Common Crawl corpus is designed to be a representative sample of text on the internet. We also tried Doc2Vec embeddings, because of results from (Baroni, 2014) that show that predictive models perform better than count models. Baroni's results are somewhat contradicted by (Levy, 2015), but Levy's work nonetheless recommends predictive models as the most reliable all-around solution as it does not perform poorly on any specific tasks. We expected Doc2Vec to also benefit from being trained on a matching domain, as the Gigaword corpus is a superset of the TAC dataset. Finally, Doc2Vec avoids the problem of averaging vectors to represent a sentence from its component words, though it is admittedly unclear exactly what information is being captured by training continuous bag-of-word (CBOW) word embeddings in combination with Paragraph IDs. (Le and Mikolov, 2014). Still, as a result of these factors, we expected Doc2Vec to yield the best performance. Our experimental setup also included removal vs. non-removal of stop-words (using spaCy's stop-word list, which is 307 words long and of unknown provenance, but is slightly less conservative than NLTK's stop-word list, which is based off (Porter, 1980)). TF-IDF naturally downweights stop-words, but other methods of embedding do not, and we hypothesized that stop-words might muddy sentence embeddings, especially GloVe embeddings, since they are averaged. Thus, we expected stop-word removal to improve performance. All embeddings are 300-dimensional.

However, as is clear from Table 7, TF-IDF beats all other representations by a very wide margin for every ROUGE measure, and stop-word removal has a negative effect on ROUGE scores for alternate embeddings. TF-IDF does not clearly beat all other representations qualitatively, as will be discussed in further sections, but its performance does suggest that further work may be well spent by optimizing the system on top of TF-IDF rather than substituting different representations for computing sentence similarity.

Surprisingly, stop-word removal led to a decline in ROUGE scores for both GloVe and Doc2Vec. The reason for this is unclear.

3.4 Redundancy Removal Experiments

LexRank was initially used for single-document summarization (Erkan and Radev, 2004). When adapted for multi-document summarization, LexRank will often select sentences that are nearly exactly the same. This is a problem for LexRank in general, since sentence salience is measured by similarity, so if an exact repetition or paraphrase exists of a very connected sentence, both are likely to be selected together, but it is greatly exacerbated in multi-document summarization, since different articles often establish the same information, update information from temporally earlier articles, or even repeat similar articles verbatim. See Appendix 6.6 for examples. Inclusion of sentences that are exact or near paraphrases is very costly in a 100 word summary, as those redundant words use up valuable space. We hypothesized that using a method to prevent selection of multiple sentences that were too similar to each other would increase Recall. SemEval-2017 had a shared task (Cer, 2017) on intra-lingual and cross-lingual semantic similarity, with systems trained on NewsWire

and tested on custom Stanford Natural Language Inference (SNLI) pairs. The winning system was an ensemble of 4 neural and 3 feature engineered machine learning systems, and the very close second (for English) was a simpler system built with WordNet sentence information content (IC). (Hill, 2016) found a Sequential-Denoising Auto-Encoder (SDAE) to be the best at paraphrase detection for the Microsoft Research Paraphrase Corpus (MSRP), with unigram TF-IDF as second best. Given this, we settled on adding maximum similarity thresholding via TF-IDF as it was already part of our system. In future it may be fruitful to combine WordNet IC with TF-IDF, as the TAC task is English only so WordNet a feasible resource.

Drawing on this work, we implemented a redundancy removal for exact or near-paraphrased sentences. We disregard the candidacy of LexRank ranked sentences that are similar to a higher ranking sentence with a similarity value above the given redundancy threshold. Results of tuning for redundancy threshold are shown in Table 8. Average recall, precision and F-score are found to be highest for redundancy threshold value of 0.5, even though there is no significant change in summary quality or even content for most cases between threshold values of 0.5 and that of 0.4 and 0.6. Comparison of Appendix 6.6 and 6.7 shows that having disregarded redundant sentences, the final summary includes a wide variety of sentences. While redundancy removal improved the content selected towards the final summary, we notice that the summary ordering is further affected and degrades post redundancy removal. Section 4 discusses methods to improve the content ordering aspect of the system.

4 Discussion

4.1 First-sentences vs. LexRank as Baseline Content Selection

As expected for the inverted pyramid structure of Newswire corpus content, the first-sentences baseline is hard to beat. It was in fact, *harder* to beat than anticipated, as LexRank did not outperform first-sentences in any area save ROUGE-1 Precision, in which the difference is not large enough to be reliably repeatable.

However, the results were qualitatively different. Many first sentences are a type of short abstract summarizing all topics that follow, but some are specific appeals to emotion or descriptions of an incident that were 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 further analyzed the sentence length and lexical diversity (defined as the ratio of unique to total words, per document) of the model summaries and the LexRank baseline summaries by performing a Mann-Whitney U test[14] on the distributions over the two summary collections. We pre-selected $p = 0.01$ and found that the difference between both distributions easily passed the significance test. The average number of words and Lexical diversity Ratio are presented in Table 10. 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.

4.2 Sentence Embeddings

Our experiments with alternate sentence embeddings trained on larger corpora were much worse than we expected, especially in light of belief that data abundance leads to better trained systems[15].

[14] https://en.wikipedia.org/wiki/Mann\%E2\%80\%93Whitney_U_test

[15] This is broadly true across many tasks, and is one of the purported benefits of the GloVe vectors of (Pennington,

	ROUGE-1			ROUGE-2			ROUGE-3			ROUGE-4		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
TF-IDF	**28.55**	**21.33**	**24.15**	**7.55**	**5.63**	**6.38**	**2.43**	**1.81**	**2.05**	**0.86**	**0.65**	**0.73**
GloVe	20.83	16.60	18.38	4.48	3.60	3.98	1.55	1.25	1.38	0.57	0.46	0.50
GloVe (no stops)	19.68	12.02	14.74	4.51	2.79	3.41	1.27	0.78	0.95	0.34	0.22	0.26
Doc2Vec	20.78	16.52	18.28	3.85	3.13	3.43	0.93	0.78	0.84	0.25	0.23	0.24
Doc2Vec (no stops)	20.58	10.48	13.61	3.42	1.78	2.30	0.95	0.52	0.66	0.24	0.14	0.18

Table 7: ROUGE results on DevTest with LexRank for different sentence embeddings before improvements made towards final system

	ROUGE-1			ROUGE-2			ROUGE-3			ROUGE-4		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
0.5	**31.92**	**20.84**	**24.77**	**8.48**	**5.71**	**6.74**	**2.79**	**1.92**	**2.26**	**1.09**	**0.79**	**0.91**
0.6	31.65	20.62	24.52	8.33	5.61	6.62	2.76	1.90	2.23	1.04	0.76	0.88
0.7	31.76	20.59	24.56	8.31	5.54	6.56	2.69	1.84	2.16	0.996	0.72	0.83
0.8	31.49	20.42	24.36	8.20	5.47	6.48	2.63	1.80	2.12	0.98	0.71	0.82
0.9	31.60	20.55	24.48	8.23	5.50	6.51	2.68	1.85	2.17	1.03	0.75	0.86

Table 8: ROUGE results on DevTest with LexRank for different redundancy removal thresholds

	ROUGE-1			ROUGE-2			ROUGE-3			ROUGE-4		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
DevTest	31.92	20.84	24.77	8.48	5.71	6.74	2.79	1.92	2.26	1.09	0.79	0.91
EvalTest	34.13	22.49	26.76	8.83	5.81	6.91	2.89	1.99	2.33	1.36	0.96	1.11

Table 9: Final System ROUGE results for TF-IDF including redundancy removal thresholding on DevTest and EvalTest with LexRank

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

Table 10: Distributional Statistics - Average number of words(**Avg Words**) per sentence and the Lexical Diversity Ratio (**LDR**) for the Model and LexRank summaries.

Unfortunately, it is actually hard to analyze *why* they performed so poorly. GloVe should capture more global context, but perhaps it is capturing the wrong global context from a very general training corpus, and the more hyper-local awareness of TF-IDF within a document cluster is superior. That could also be a reason for Doc2Vec's weak performance, as neither represents the specific context of the cluster being summarized. An area for future work could be to control variables to be better able to compare the different embeddings. That said, (Levy, 2015) experimented with nine hyperparameter settings for five types of embeddings, concluding that the individual choice of model matters less than the hyperparameter tuning. In this light, it seems prudent to optimize for one embedding type.

Despite the weak ROUGE scores, we did notice qualitative improvements in Doc2Vec, discussed below. The ability to extend LexRank to new sentence representations is also valuable for further research and may be necessary for representing smaller chunks of sentences for further redundancy reduction.

While Chronological sentence ordering adds some coherence to the discourse, we observe that spaCy GloVe vector based system generates a summary wherein an incomplete sentence features as the second sentence in the summary. Doc2Vec based systems add to the summary quality over spaCy's GloVe based systems which tend to grant higher salience to quoted sentences and phrases. Doc2Vec summaries often end with a concluding statement, which is positive, but take a leap in discourse, therefore breaking the continuity of the summary. These examples (Section 6) highlight the major weaknesses of the current system, namely some extent of discourse incoherence, severe lack of anaphora and coreference resolution.

2014), which are faster to train and would therefore get performance boosts from consuming additional data

A possible solution to these problems could be to make use of discourse and semantic features. It is worth noting that incorporating discourse features would require a mostly supervised system. This could prove to be tedious and non-feasible since human annotated discourse features are not scalable for a large dataset. Furthermore, the feasibility and pay-off of incorporating semantic features remains to be explored given that typically semantic features are not very helpful towards selecting summaries.

4.3 Redundancy Removal and Sentence Normalization

It was noted that the summaries generally contain longer sentences, but we were unable to resolve this bias successfully. There may be more methods of sentence length normalization to try. There may also not be many informative short sentences available, since LexRank treats a sentence as an independent unit, and shorter sentences are more dependent on inter-sentential relations. The use of distributional semantics to restrict synonymy and sentence similarity improved performance quantitatively and qualitatively, as expected. The most promising future improvement in this area is likely to be to combine sentence compression with redundancy removal, which would allow long sentences but remove both non-contentful clauses and phrases, and remove already included phrases. Methods of event extraction could also be used to ensure only one sentence per major event. There was a SemEval-2013 shared task (UzZaman et al., 2013) on a TimeBank + AQUAINT corpus (a superset of TAC) where many feature-engineered systems achieved F1 scores exceeding 75 at identifying events and classifying them into types. Event information could also be used to improve the ordering of the summary content. In light of this, event extraction is a feasible next step.

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 a summarization of a murder, all the details of the actual murder are missing (see Section 6). 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 bite-sized pieces should have strong objectives for concept and/or event recall. Event extraction, as mentioned in the context of redundancy removal, could also be used as an evaluation metric to penalize poor event recall. This will be a necessary metric to incorporate into future efforts to ensure that summaries are representative of the documents that they summarize. It is also algorithm independent and could be used for any TAC summarization system, not just for LexRank.

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. This was found to not be the case for our data as the first-sentence baseline outperformed all other baselines for nearly all ROUGE metrics.

Because of LexRank's inherent proclivity to choose long sentences, we believed that normalizing sentences by their length would regularize similarity when choosing nodes in the graph, but found this to not only penalize long sentences, but also to prefer incredibly short and non-salient sentences.

We further hypothesized that using complex sentence representations via neural and distributional semantic embeddings trained on large corpora would improve LexRank sentence saliency, and this was also shown to not be the case, with TF-IDF outperforming all other variants.

Extensive experimentation with the system indicates that summary quality could be improved upon by improving discourse coherence, anaphora resolution and penalizing redundancies to minimize paraphrasing of the same salient topics. Additionally, modeling topic relevance in LexRank scores and removal of bias towards long sentences in output rather than in selection stage could generate more content rich and representative summaries. Finally, the summary ordering could be improved upon by incorporating event information in the information ordering module.

References

Baroni, M., Dinu, G., and Kruszewski, G 2014. *Don't count, predict! A systematic comparison of context-*

counting vs. context-predicting semantic vectors. In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers) (Vol. 1, pp. 238-247).

Bollegala, D., Okazaki, N., and Ishizuka, M. 2012. *A Preference Learning Approach to Sentence Ordering for Multi-document Summarization.* Information Sciences 217, 22, 78-95.

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

Cer, D., Diab, M., Agirre, E., Lopez-Gazpio, I., and Specia, L. 2017. *SemEval-2017 Task 1: Semantic Textual Similarity-Multilingual and Cross-lingual Focused Evaluation.* arXiv preprint arXiv:1708.00055.

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.

Hill, F., Cho, K., and Korhonen, A. 2016. *Learning distributed representations of sentences from unlabelled data.* arXiv preprint arXiv:1602.03483.

Kiss, T., and Strunk, J. 2006 *Unsupervised multilingual sentence boundary detection.* Computational Linguistics, 32(4), 485-525

Le, Q., and Mikolov, T. 2014 *Distributed representations of sentences and documents.* In International Conference on Machine Learning (pp. 1188-1196).

Levy, O., Goldberg, Y., and Dagan, I. 2015 *Improving distributional similarity with lessons learned from word embeddings.* Transactions of the Association for Computational Linguistics, 3, 211-225.

Pennington, J., Socher, R., and Manning, C. 2014. *Glove: Global vectors for word representation* In Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP) (pp. 1532-1543).

Porter, M. F. 1980 *An algorithm for suffix stripping.* Program, 14(3), 130-137.

UzZaman, N., Llorens, H., Derczynski, L., Allen, J., Verhagen, M., and Pustejovsky, J. 2013. *Semeval-2013 task 1: Tempeval-3: Evaluating time expressions, events, and temporal relations.* In Second Joint Conference on Lexical and Computational Semantics (* SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013) (Vol. 2, pp. 1-9).

6 Appendix

6.1 TF-IDF 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.

6.2 Doc2Vec without stopword removal

When they came in February, shortly after their son's killing, they attained near-celebrity status _ trailed by television cameras, greeted by crowds and transported from place to place by motorcade.

What happens next could intensify, rather than quell, the racial tensions that have been simmering in the two months since the shooting.

6.3 SpaCy without stopword removal

That case is set to begin on Tuesday.

and another falling down some steps as if he had been shot.

Friday, acting Justice Patricia Anne Williams, who refused defense requests to put off scheduling a date for the case, chose the Jan. 3 date after the defense lawyers said they would need time to go through the prosecution's evidence and discuss possible pretrial motions.

It was a point that Worth disputed outside court by saying, "I don't think I have a conflict of interest, though it is obvious that Kornberg would like for me to go."

6.4 Doc2vec with stopword removal

The lawyers also said there could be pretrial scheduling conflicts because of other trials some of them are involved in, like the Abner Louima case, in which officers are accused of brutalizing a Haitian immigrant.

Police officials declined to say where the officers _ Kenneth Boss, Richard Murphy, Carroll and

McMellon _ would be working, adding only that they were on modified duty and carrying no guns or badges.
They left the courtroom to the cheers of nearly 200 other officers, gathered outside the Bronx County Building.

6.5 SpaCy with stopword removal

"We are waiting for this, for justice," he said, "and we feel strong."
The four officers fired 41 shots, hitting Diallo 19 times.
The Rev. Al Sharpton, who organized the daily protests over the Diallo shooting and is an adviser to the family, also is expected to attend the meeting, said Steven Reed, Johnson's spokesman.

6.6 TF-IDF example of redundant sentence selections

All of the below sentences were selected for inclusion in one summary, before redundancy removal was implemented. They are all from different news stories, and, despite differences in details, contain nearly the same content. It in unclear whether the high level of verbatim overlap is due to one reporter reusing content, different reporters reusing each other's content, or to a translator reusing their translations (as this example was translated). We found this high level of exact overlap amongst many of the examples.

PORT MORESBY, Papua New Guinea (AP) A tsunami spawned by a 7.0 magnitude earthquake crashed into Papua New Guinea's north coast, crushing villages and leaving hundreds missing, officials said Sunday.
PORT MORESBY, Papua New Guinea (AP) A tsunami spawned by a 7.0 magnitude earthquake crashed into Papua New Guinea's north coast, crushing villages and killing nearly 600 people, officials said Sunday.
PORT MORESBY, Papua New Guinea (AP) Huge sea waves set off by an earthquake crashed against Papua New Guinea's north coast, killing at least 70 people and crushing villages, the country's National Disaster Center said.

6.7 Sample summary After redundancy removal for TF-IDF embeddings

Summary generated for devtest after redundancy removal using redundancy threshold of 0.5:

It's complete devastation."
UR; MORE
"They're dead . . .
The 10-metre tsunami engulfed the heavily populated villages near Aitape, 800 kilometers north of Port Moresby on Friday night - 30 minutes after an undersea earthquake of about 7.0 on the Richter scale in the area.
PORT MORESBY, Papua New Guinea (AP) _ A tsunami spawned by a 7.0 magnitude earthquake crashed into Papua New Guinea's north coast, crushing villages and killing nearly 600 people, officials said Sunday.

6.8 Sample summary generated by final system on EvalTest

Nichols said police were seeking other suspects.
Simpson is expected to be questioned again Sunday and may face arrest, police Lt.
The charges against Simpson will include robbery with a deadly weapon, conspiracy to commit robbery and burglary with a firearm, all felonies, Dillon said.
A man was arrested in connection with an alleged armed robbery of sports memorabilia involving O.J. Simpson, and police said Sunday the former football star may be arrested as well.