Deliverable #3: Summarization Improvement; Information Ordering

Paige Finkelstein **Jacob Hoffman** **Wesley Rose** **Joshua Tanner**
University of Washington
Department of Linguistics
{plfink, jrhoff, warose91, jotanner}@uw.edu

Abstract

In this deliverable report we explicate the work we've done to improve our baseline system from deliverable #2. We have improved our content selection method by incorporating topic-focused summarization via bias. We have also refined our information ordering approach by making our temporal ordering considerations more robust and by adding succession likelihood and topical closeness "experts", inspired by the approach outlined in Bollegala et al. (2012). Finally, we have improved our content realization by adding a number of heuristics to filter out less central content.

1 Introduction

We have worked to improve our multi-document summarization system that produces brief summaries from topic-oriented document clusters of articles. For this deliverable, we have focused on further refining our content selection approach and adding a robust information ordering method. We have also expanded our content realization strategy and made architectural and configuration improvements to our system.

Our new system implementation achieves an average ROUGE-2 recall score of 0.08042, improving on our previous system's ROUGE-2 recall score of 0.06145. While our expanded information ordering approach does not contribute to higher ROUGE scores as the content selection and content realization improvements do, we believe that it does result in more comprehensible summaries overall.

2 System Overview

Our system architecture remains largely the same as explicated in deliverable #1. In this deliverable report, we will explain only the additions and improvements, and it should be assumed that everything else remains as outlined in our previous report. In particular, our modules that provide functionality for data parsing, loading, storing, and re-loading have been continuing to work effectively and have not required any additional work. We have continued to identify some patterns for strictly unnecessary text artifacts to strip out during the pre-processing stage, but, otherwise, our pre-processing module remains largely unchanged.

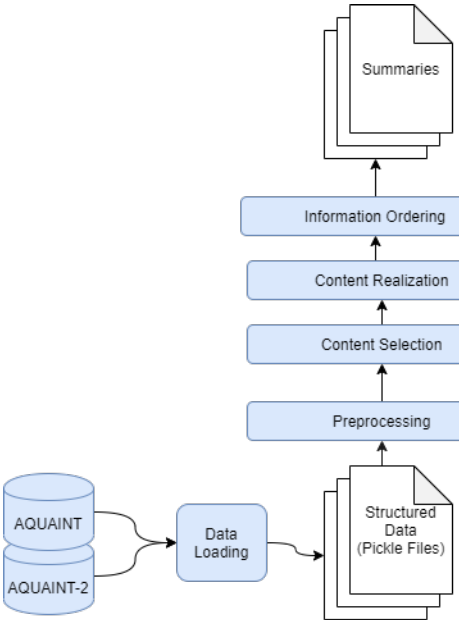

Figure 1: Updated system architecture diagram

The one significant architectural change that has occurred since our last deliverable report is that we decided to switch the order of our information ordering and content realization modules in the summarization pipeline. As Figure 1 demonstrates, content realization now occurs before information ordering. We made this decision because in the course of implementing our more robust information ordering method, we determined that the most effective way to approach ordering was to be able to treat it as an isolated problem. Before, this was impossible because the content realization module that occurred after information ordering was cutting further segments from the ordered sentences and was responsible for limiting the summary to 100 words or fewer. As a result, the information ordering module had to be cognizant of keeping the most "information rich" sentences near the beginning of the summary in order to be sure that they would not be cut when content realization pared down to 100 words.

In the new architecture, however, the information ordering module now receives the realized sentences exactly as they will be kept in the final summary output and can therefore focus solely on the problem of trying to order the sentences in the most effective manner.

3 Approach

3.1 Content Selection

For D3 we tried multiple methods in content selection: two different methods incorporating our heuristic score and LexRank.

3.1.1 LexRank

For this deliverable, we completed an implementation from scratch of LexRank as explained in Erkan and Radev (2004). Furthermore, we tried several methods of expanding upon LexRank, but with little success. We tried using an average of spaCy's word vector similarity as the measure of link strength in graphs, which saw a nearly 50% drop in performance, suggesting that this is not a good substitute for the modified-idf-cosine mentioned in LexRank. We also tried an idf-weighted average of word vector for similarity, inspired by the idf-modified-cosine method, which achieved 80-90% of the performance of the original LexRank method. This clearly demonstrates that idf plays a critical role in successfully building the graphs that power LexRank.

In an effort to query-focus our summaries, we also experiemented with Biased LexRank (Otterbacher et al., 2008). We used the first biasing method mentioned in the paper based on idf-weighted word overlap, originally from Allan et al. (2003). We found that the biasing improved performance slightly, between 5-10% depending on run configuration. Additionally, we experimented with the word probability based models for biasing and graph-building mentioned in Otterbacher et al. (2008), but found these provided little to no performance improvement and ran substantially slower.

It is unclear why LexRank based methods are not performing as well for us as for the original authors, though it is worth noting that the papers are very vague about the finer details of some of their implementations, so it is very possible our implementations differ from theirs in some way we have been unable to identify. In particular, despite lemmatization not being mentioned in either the original LexRank paper or the Biased LexRank paper, an example in the Biased LexRank paper clearly assumes lemmatization as part of the process. However, both lemmatization and stopword removal have provided only marginal gains. More importantly, LexRank performance peaked at a ROUGE-2 recall score of approximately 0.07. Given that its performance has consistently lagged behind our n-gram based approach, we are abandoning further work on it except as a possible feature in a classifier to be trained later.

3.2 Heuristic Score

As mentioned in D2, we used a weighted average of different n-gram probabilities as well as headline NER overlap to derive a score. We experimented with two new components to this score, and although they perform reasonably well when used alone, our system did not seem to benefit from these two components.

1. **Cartesian Pair Score:** Inspired by the generation probabilities in Otterbacher et al. (2008), this score is composed of the average probability that two words occur in the same sentence. First, we count the unique pairs of words for each sentence for a document group. Then, we evaluate a sentence's score by finding the average probability that its words occur in the same sentence across the document.

2. **Query Words Score:** This score is simply derived by how many words in the sentence, occur in the query. This score showed very little effect on ROGUE-2, unless taken with an extremely high weight.

We've found that using only weighted unigram and bigram averages work best at this point. All scores have weights which are manually tuned, and all components ignore stop words.

3.3 Heuristic Selection Methods

3.3.1 Basic Per-Article Selection

First, we compute the relevant n-gram probabilities across the document group. Then we consider one article at a time, selecting the top-N (configurable) sentences to content ordering. For each word in each of the of the top-N sentences, we re-weight the n-gram probability distribution. We do this the same way unigram probabilities are reweighted in Vanderwende et al. (2015). That is, $P(unigram)_{new} = P(unigram)^2_{old}$. Reweighting bigrams yields reduced performance on both AQUAINT and AQUAINT-2 datasets, so we opted not to do so. Everything except reweighting and configurable number of sentences choses is the same as in D2 for this method.

3.3.2 Glob Selection

'Glob' refers to treating all of the articles as one block of text. Unlike Per-Article selection, the order in which articles appear in the document group has no effect on the outcome. We simply compute heuristic scores for every sentence, choose the best one, and then re-weight the unigram distribution as we do in Per-Article selection. We also experimented with weighting the normalized glob scores with query biases produced by our LexRank implementation. We found some evidence that incorporating query bias improved scores, but not by much. Additionally, Glob seems to be more consistent than Per-Article selection across datasets. When incorporating the bias, we calculate the final score as $Final = (1 - Bias\_Weight) * Heuristic\_Score + (Bias\_Weight) * Bias\_Score$.

3.4 Information Ordering

In our base implementation of the information ordering strategy, we took some initial inspiration from the notion of a "preference learning" method outlined in Bollegala et al. (2012). For this deliverable, we expanded on this approach by making our *chronological expert* and the *topical-closeness expert* more robust and by adding in a third expert based on *succession likelihood*, which was also proposed in Bollegala et al.

For chronological ordering in our updated system, we moved from Bollegala et al.'s straightforward paradigm whereby article publication date is given strict preference in determining order and followed secondarily by sentence order in the document, to a more complicated schema in which article publication date as well as the index position of the sentence in the specific article are given weighted values to contribute to the overall preference score. Furthermore, sentences that occur at index positions 0 or 1 in their article (if any exist in the summary text) are given an even greater weight when selecting the starting sentence for the summary.

For topical closeness in our previous system, we were using a comparison of cosine similarity between word vectors to identify redundant sentences to move to the bottom of our ordering where they would likely be cut out in the content realization when the summary was limited to 100 words. With our new architecture, described in section 2 above, this is no longer necessary. We instead now use the topical closeness expert to contribute to the preference score in selecting which sentence comes next. We are still using built-in spaCy functionality to compare an average of the word vectors to determine cosine similarity, and we experimented with different values to determine how heavily to weight this expert in the overall calculation for preference.

Finally, for the new succession likelihood expert, we are using the Transformers library (Wolf et al., 2019) to access the BERT pre-trained language model (Devlin et al., 2018) to generate probabilities about the likelihood of one sentence following another. As next sentence prediction was one of the two initially training measures for BERT, we thought it seemed like a promising candidate for the succession expert. However, we discovered that information ordering results that give significant weight to the succession expert seem to be mixed in terms of improving readability, and so we provided functionality to exclude this 'expert' from the weight calculation when specified.

3.5 Content Realization

For deliverable 3, we added and modified a number of content realization features, however we have not settled on the best way to include these features in the context of the full pipeline. All of the features in realization are configurable. Some features are binary, and can simply be switched on or off, and some have numeric values. Many of these features interact with the other modules in complex ways, and more thorough experimentation is needed to find the optimal combination. In this section we will present each feature with its configurable values, as well as the congfiguration that was used to achieve the score that we are reporting for D3.

The redundancy check has been moved from information ordering to content realization. Inspired by recent trends in NLP, we developed a method to apply BERT to redundancy checking. The HuggingFace Transformer library (Wolf et al. (2019)) provides a BERT model that has been fine-tuned on the Microsoft Research Paraphrase Corpus (Dolan and Brockett (2005)) to detect paraphrases. We thought that paraphrase likelihood was a good stand-in for redundancy likelihood, so we implemented this into our pipeline along with a configurable threshold (0-1) for what to label as redundant. Disappointingly, we found that this method was not only much slower, but it also failed to improve our metrics. For this reason, we default to using spaCy's vector similarity measure with a configurable threshold (set to 0.97 for our reported scores). We find it interesting that our best results are obtained with such a high threshold, but we attribute this to our targeted content selection algorithm.

We implemented a few heuristics that are used to strip out full sentences which can each be switched on or off. Three very simple heuristics are filtering out any sentences containing quotes, sentences with question marks, and sentences that do not meet a minimum length requirement. Finally, we attempt to filter out ungrammatical sentence fragments using spaCy's built-in dependency parsing to flag sentences that do not have a nominal subject. For the results reported in this deliverable, quote and question removal were turned on, minimum length was set to 4, and the ungrammatical filter was turned off.

We also implemented two regex based heuristics for realization. We did not use either of these methods to achieve our reported results, but we will continue to experiment with them going forward. Both methods are configurable based on lists of regex expressions in the configuration file. One method will remove the full sentence whenever the sentences matches a regex in the list. The second method will remove only the subsequence within the sentence that matches the given regex. The main benefit to this is to easily experiment with new filters, and any valuable filters that we find may be hardcoded in the future.

Finally, we have improved two features that were included in D2, namely, sentence-initial adverb removal and appositive removal. We have lessened the aggressiveness of the adverb removal to reduce the number of ungrammatical sentences created, and we have improved the rules of the appositive remover to leave us with fewer undesirable artifacts such as conjunctions, punctuations, and whitespace.

4 Results

The weights for Per-Article Selection and Glob Selection heuristic scores were all $unigram\_weight = 0.2, bigram\_weight = 0.9$. The bias weights for Glob are shown in Table 1.

Per-Article selection achieved the highest overall ROGUE-2 recall scores on both the data sets. Selecting 2 sentences per article yielded higher scores than our other methods except for the recall score on the AQUAINT-2 dataset, for which 1 sentence per article yielded the highest score. With 1 sentence per article, our system achieved 0.07306 on AQUAINT and 0.08032 on AQUAINT-2. With 2 sentences per article, it achieved 0.08042 on AQUAINT and 0.07713 on AQUAINT-2. Choosing 2 sentences per article seemed much more consistent across the datasets as the difference between AQUAINT and AQUAINT-2 recall scores was -4.27%. This difference increases to 9.04% when only 1 sentence is selected per article.

The scores for Glob (bias and basic) were all very similar on the AQUAINT set. However, there was more variance in the scores for the AQUAINT-2 data set, which changed substantially even with a 0.05 increase in the bias weight. Scores for the bias methods can be seen in Table 1.

LexRank performed the worst on the AQUAINT set with a score of 0.06882, but it did outperform basic Glob selection and bias Glob selection (bias_weight=0.01) on the AQUAINT-2 data.

Table 1: D3 Methods Comparison

Method	**ROUGE-2**		
	AQUAINT	**AQUAINT2**	**Diff**
	R		**%**
Per_Article (1 sent-per)	0.07306	**0.08032**	9.04%
Per_Article (2 sents-per)	**0.08042**	0.07713	-4.27%
Glob	0.07810	0.07330	-6.55%
Bias Glob (w=0.1)	0.07793	0.07445	-4.67%
Bias Glob (w=0.05)	0.07812	0.07558	-3.36%
Biased Lexrank	0.06882	0.07488	8.09%

Table 2: Average ROUGE Recall Scores for D2 and D3 on the AQUAINT data set

Deliverable	ROUGE-1 R	ROUGE-2 R
D2	0.21649	0.06145
D3	0.25856	0.08042

As shown in Table 2, our changes from our D2 system to our D3 system have resulted in a 0.01897 improvement in ROUGE-2 recall micro-averaged score.

5 Discussion

5.1 Error Analysis

Despite the gains we have obtained on our score between D2 and D3, we believe there is still substantial room for improvement. There are a few specific trends in errors that we think will be straightforward to address, as well as room remaining for improved content selection through additional experimentation.

One trend in errors is the inclusion of ungrammatical sentence fragments, such as *A look at the situation of seniors and others at a now-closed Columbine High.* and *The people knowledgeable about the case said.* We can think of a few ways of addressing the inclusion of these fragments. We have begun working on a feature in content realization to filter out ungrammatical sentences by using spaCy's dependency parser. Our current idea is to remove any sentences that don't have a nominal subject. This looks like a promising method, but more experimentation is required. Another possible approach is to consider why the fragments are coming through in the first place. In some cases, the problem is introduced when spaCy splits paragraphs into sentences, such as splitting the following input into two sentences before and after "Gun ": *At that point, Carroll yelled, "Gun!" the people knowledgeable about the case said.* For our next deliverable, we will consider different methods of splitting the input into sentences.

Another problem we found was unnatural and confusing use of referents, such as using pronouns in the wrong place and using full descriptors of entities more times than necessary. More specifically, there are many instances of the pronoun *they* appearing without a clear co-referent, and also many examples of the same entity being fully named in two back-to-back sentences. Luckily, there are multiple papers with ideas that we can use to address these problems, and we will experiment with them for our next deliverable.

Finally, some of the trimming that is done in the content realization module is still not perfect. For example, consider the transformation of the following sentence: *They also claim that the measure will make California, already the United States' leading hub for biotech industries, a world hub of stem cell research too.* Our system currently changes this to be: *They also claim that the measure will make California the United biotech industries stem cell research.* Clearly, content is not removed from this sentence the way it should be. We have two ideas for further work in this area: we will either try to find out concretely what is going wrong with our current parsing-based approach and fix it, or we will go the route of Conroy et al. (2006) and use a much more targeted regex-based removal method so that we can be more conservative with how we trim.

5.2 Information Ordering Updates

The additions and changes that we experimented with while working on our more robust information ordering approach had little impact on the ROUGE score metrics. This makes sense, however, as we are now treating information ordering as an isolated task in which no changes are made to the content that is included in the summary and strictly ordering changes are enacted in this module. While the ordering of sentences can marginally impact ROUGE-2 scores by changing bi-gram overlap between adjacent sentences, we did not find the difference in scores to be an effective measure for ordering success when manually inspecting the output summaries.

Our new ordering approach seems to be especially helpful in selecting a more suitable initial sentence for the summary. For example, without applying our new ordering scheme we output the following summary:

Some officials said 20 to 30 people were still missing. President Clinton said today the United States has an interest in resolving the Kosovo conflict. In New York Wednesday, the Dow Jones industrial average closed down 144.75 at 9,399.67. The latest torrent of snow smashed into the town of Valzur, where two people were killed and two survivors were pulled from the snow and debris. At least eight people have been confirmed killed and 30 others injured as two avalanches roared *through a ski resort in Austria's western state of Triol on Tuesday evening, the Austrian news media reported.*

Having the summary begin with *"Some officials said 20 to 30 people were still missing"* is obviously not ideal, as this statement depends upon prior knowledge of an event that caused people to go missing. With our new multi-faceted preference based ordering approach applied, however, we output a summary that begins with the following sentence instead: *At least eight people have been confirmed killed and 30 others injured as two avalanches roared through a ski resort in Austria's western state of Triol on Tuesday evening, the Austrian news media reported.* We observe a trend throughout the new summaries of more effectively contextualized introductory sentences, which we think is a definite improvement for readability.

However, there are still a range of cases in which our new ordering approach does not catch ordering oddities as well as we would like. In particular, there are still cases of sentences with pronoun subjects occurring before sentences in which their antecedent has been defined. One possible approach to this specific issue would be to try to implement some hard-coded rules around pronoun usage to supplement our current weighted "expert" scoring system.

6 Conclusion

In this deliverable we have focused on improving our baseline system from deliverable #2, especially in terms of enhancing our content selection approach and adding a robust information ordering method. We also made improvements to our content realization approach, which we plan to further develop in our next deliverable. We saw significant ROUGE-2 score improvements as a result of these system updates, and also believe that our information ordering changes have contributed to more readable summary outputs.

References

James Allan, Courtney Wade, and Alvaro Bolivar. 2003. Retrieval and novelty detection at the sentence level. In *SIGIR '03*.

Danushka Bollegala, Naoaki Okazaki, and Mitsuru Ishizuka. 2012. A preference learning approach to sentence ordering for multi-document summarization. *Inf. Sci.*, 217:78–95.

John M Conroy, Judith D Schlesinger, Dianne P O'Leary, and Jade Goldstein. 2006. Back to basics: Classy 2006. In *Proceedings of DUC*, volume 6, page 150.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*.

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

Jahna Otterbacher, Gunes Erkan, and Dragomir R. Radev. 2008. Biased lexrank: Passage retrieval using random walks with question-based priors.

Lucy Vanderwende, Arul Menezes, and Chris Quirk. 2015. An AMR parser for English, French, German, Spanish and Japanese and a new AMR-annotated corpus. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Demonstrations*, pages 26–30, Denver, Colorado. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing.