Ling 573 - NLP Systems and Applications

Elena Khasanova, Erica Gardner, Saumya Shah, Sophia Chan and **Vikash Kumar**
Department of Linguistics
University of Washington, Seattle
{ekhas1, erstgard, saumyahs, schan2, vikas134}@uw.edu

Abstract

Text summarization is the process of distilling the most important information from a source to produce an abridged version for a particular user and task. In this paper, we discuss a method used for extractive multi-document text summarization. We use topic modelling to find the sentences in a document that are most relevant to a topic. Thus we rank the most relevant words in a topic and uses it to compute a phi score or a word-topic score. The phi score allows us to pick out a sentence that strongly represents the topics in the document. Our basic end-to-end system produces a ROUGE-1 average F-1 score of 0.21316 and a ROUGE-2 average F-1 score of 0.05031. Our system performs nearly as well as the baselines, and we find no statistically significant difference.

1 Introduction

Automatic Text Summarization is one of the most challenging and interesting problems in the field of Natural Language Processing (NLP). The demand for automatic text summarization systems is spiking these days thanks to the availability of large amounts of textual data. There are many different paths from a set of documents to a summary.

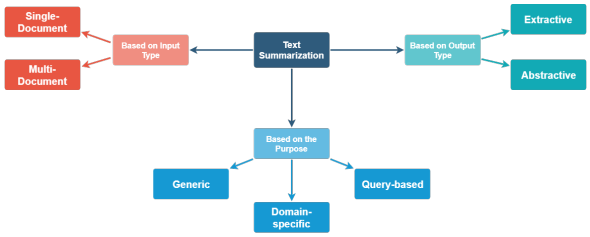

Figure 1: Types of summarization methods

Figure 1 shows the different types of summarization methods based on the type of input, type of output and purpose. Single Document summarization occurs where the input length is short. Many of the early summarization systems dealt with single document summarization. Multi Document is where the input can be arbitrarily long and the summarization spans many documents.

Based on the purpose, generic occurs when the model makes no assumptions about the domain or content of the text to be summarized and treats all inputs as homogeneous, domain-specific summarization occurs when the model uses domain-specific knowledge to form a more accurate summary. For example, summarizing research papers of a specific domain, biomedical documents, etc and query-based summarization takes place when the summary only contains information which answers natural language questions about the input text.

Based on output type, we have extractive summarization, where important sentences are selected from the input text to form a summary. Most summarization approaches today are extractive in nature. We also have abstractive summarization, where the model forms its own phrases and sentences to offer a more coherent summary, like what a human would generate.

For the deliverable 2, we have decided to explore the use of topic modelling in topic-wise summarization. The focus of this deliverable lies on the content selection, and we have experimented with different preprocessing methods which will best complement the LDA algorithm and help it produce the best results.

2 System Overview

Our approach highlights a multi-document generic extractive summarization system.

We first get the document sets from the topics using XML readers and organize them to be fed into the preprocessing module. The preprocessing module reads all the documents by topic. The

main steps involve stop word removal, lemmatization and sentence segmentation and tokenization.

To perform summarization, we use sentence ranking using topic modelling. Topic modelling is an unsupervised method of finding topic clusters within a document and assign a set of words to each topic. Based on the ranking of these words we can find the sentences that contribute most to the topic and hence select content for the summary. Thus the lemmatized sentences for each document provided as input are then organized into topics with a set of vocabulry for each word. Using the word-topic score we find the sentences that are most relevant to summary and is then passed on to the information ordering module.

In the information ordering module, we selected the sentences based on the LDA score from the content selection output and picked the highest ranked sentences to go first.

For content realization, we experimented with a couple heuristics to get to the 100 word limit such as eliminating sentences shorter than 8 words, removing parenthetical expressions, removing adverbs.

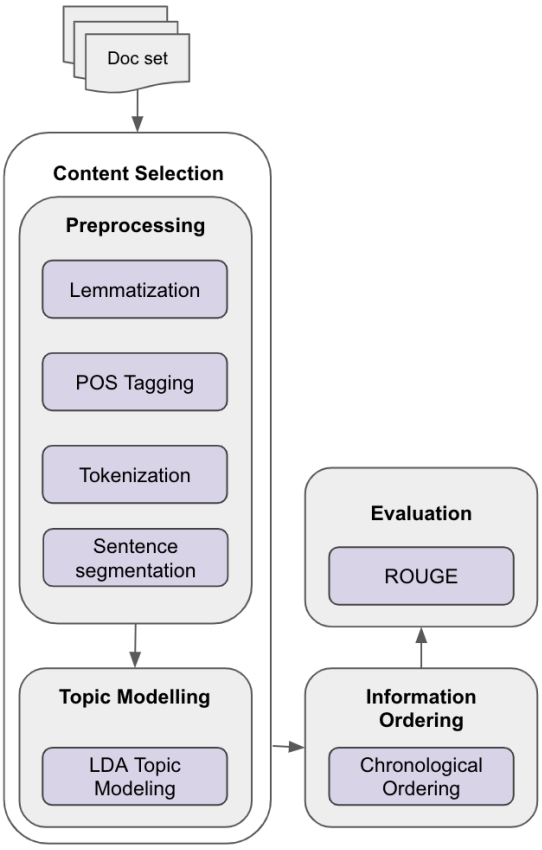

Figure 2: Our proposed system architecture

3 Approach

3.1 Data Extraction

To extract the train, test and development files, we use an XML reader called BeautifulSoup [1], developed in Python. We organized the data into a dictionary with topic_id as the key and the set of documents in the topic as its value. We have also retained parameters like date time, title and narrative, which may provide a good basis for our summarization system.

3.2 Content Selection

The two main components in our selection module includes the preprocessing stage and the topic modelling component using LDA.

3.2.1 Preprocessing

The preprocessing stage was motivated by the main content selection method - topic modeling with Latent Dirichlet Allocation (LDA), which benefits from as little noise as possible in the input data. Other direct consumers of the preprocessing output - information ordering and content realization - required preserving all relevant inpur information. We used SpaCy[2] processing pipeline to obtain various properties of the sentences.

Our preprocessing for topic modeling simply consists of the following steps:

- stopwords removal;
- lowercasing and lemmatization;
- sentence segmentation and tokenization.

This ensures that important topical concepts are not obscured by morphology and increase the performance of LDA. We considered another alternative to simple lemmatization - *delexicalization* of the topical words after lemmatization using vector similarity scoring, which would take advantage of grouping synonymous concepts under the same lexical item. This component was not eventually selected for the first deliverable as it required more careful experimentation and validation to ensure its successful interplay with LDA model.

To retain as much information as possible to further used in downstream tasks, we summarized the properties and various annotations and statistics of the input sentences and grouped them by the

[1] https://www.crummy.com/software/BeautifulSoup/
[2] https://spacy.io/

topic (document set) identifier. At this stage, we filtered out *interrogative* sentences as those that would be less likely to represent the main content of the news articles. The annotations and statistics include *named entity recognition, part-of-speech tagging, dependency parsing, sentence lengths, sentence index, document index,* and *total number of sentences* in corresponding documents. The indices and lengths are useful in information ordering and content realization, while POS-tagging, dependency parse and named entity recognition are assumed to serve as heuristics and help take advantage of the properties of news genre and weight the candidate sentences accordingly: a good news article normally answers such questions as what, when, and where happened and who were the participants. Some of this information was removed from this submission for clarity and consistency but will make it back as we improve the system. 3 gives the overview of this component.

3.2.2 Topic Modeling

Latent Dirichlet Allocation(LDA) is a generative statistical modeling technique that allows us to estimate unobserved (latent) topics from an observed distribution of documents (bag of words). The goal of LDA is to best estimate the topics that could have generated the observed documents. Each document is described as a distribution over a fixed number of topics and each topic is described as a distribution over the known vocabulary. For our purposes, we build an LDA model with three topics and used the most relevant terms of each topics as a measure of word relevance. We have used the *gensim* package which provides both singlecore and multicore(parallelized) implementation of the LDA model. We have used the default value of $\{1/\text{number of topics}\}$ for the hyperparameter *alpha* .

Since LDA gives us two probability distributions, we are using the word-topic distribution to estimate the relative importance of each sentence. The idea here is to use the computed *phi* score to assign a rank to each sentence in the document set based on the number of topics this sentence covers and pick out sentences that most strongly represent the discovered topics in the given document set. We intend to use more sophisticated methods to generate candidate sentences by increasing the number of topics hyperparameter and measuring *topic coherence* to discard irrelevant topics, this method might help us in selecting the most semantically relevant sentences.

3.2.3 Alternatives and Improvements

LDA gave us reliable scores to select the most content rich sentences based on the latent topics distribution over sets of related documents. We considered other metrics to score the sentences that are known to be performant for extractive summarization such as TF-IDF and concreteness scores. Boths methods were implemented but kept for the future iterations of our system. TF-IDF (term frequency-inverse document frequency), which reflects the importance of a lexical unit to a document in comparison to other documents. This score was computed using *TfidfVectorizer* from *sklearn* module. Concreteness scores are obtained from the dataset created by (Brysbaert et al., 2014). This dataset contains the concreteness ratings for 40 thousand English lemma words collected in a behavioral experiment with Amazon Mechanical Turk. This dataset is claimed to represent all American English words known to 85% of the raters and can therefore sufficiently cover the content of the news articles. The scores were summed for each sentence. This metric can be a good proxy for selecting the candidate sentences specifically from the news articles since we expect the relevant information to be written in accessible and concrete language. We assume that the three metrics - LDA scores that rank the underlying content, TF-IDF scores that rate the representative power of vocabulary items, and concreteness scoring that works in line with the genre - will work better in combination. For the next submission, we are aiming to find the right ratio of the three metrics to refine the scoring of the candidate sentences and investigate the effects of delexicalization on the system performance.

3.3 Information Ordering

We explored several strategies for information ordering, and ultimately settled on an approach that built on our content selection methodology. Through topic modeling, we were able to assign an LDA rank to each sentence based on how well it represented the topics of the document set.

Originally we set out to use chronological ordering as a metric for ordering information but we decided to abandon this approach after seeing the poor performance of chronological ordering methods in comparison with other options. This deci-

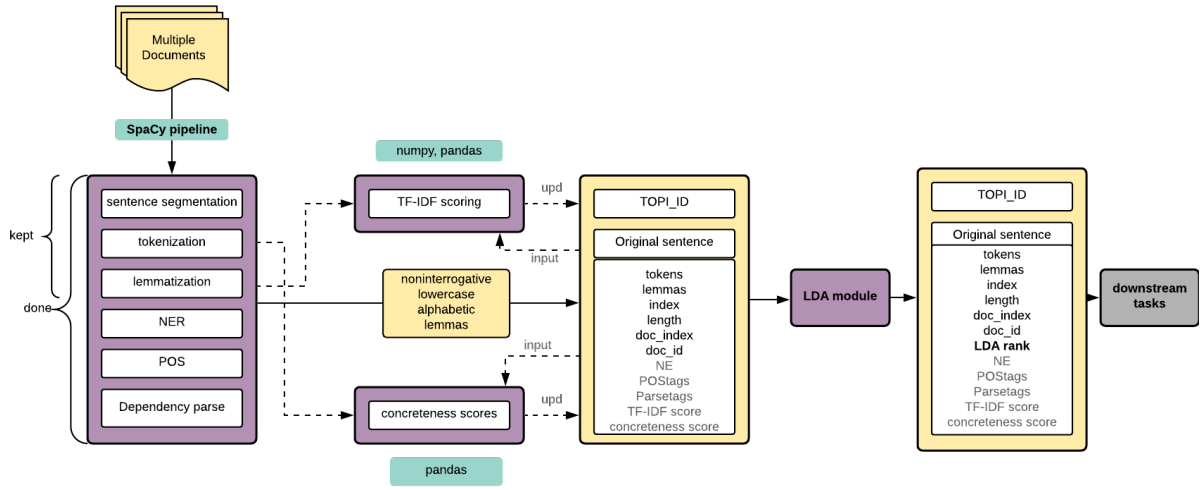

Figure 3: The preprocessing module

sion was motivated by the findings of Barzilay et al., in which the chronological ordering approach gets the worst scores in human evaluation as compared to majority and augmented ordering (Barzilay et al., 2002). Their best-performing strategy was the augmented algorithm, which combined chronological ordering based on publication date with grouping by theme. Along these lines, we plan to incorporate chronological ordering for the next deliverable, and will use this in conjunction with our existing metric. We a lso plan to incorporate topicality by using cosine similarity to add sentences most similar to those already in summary. This approach, as discussed in lecture, will promote cohesion in our output summaries.

3.4 Content Realization

Since content selection was the focus of this deliverable, we did not have time to implement all of our planned content realization strategies.

For this deliverable, we incorporated several basic heuristics to cut down on unnecessary information and do sentence compression. The purpose of this was to free up space in the summary for more useful information. Since we were using an extractive approach, we added information to the summaries sentence by sentence, but cutting out unnecessary information meant we were able to incorporate additional sentences.

Our first heuristic was to ignore sentences with fewer than 8 words. We assumed these would by bylines or segments of titles, so would be less likely to contain pertinent information. We also removed some parenthetical expressions, in particular, those enclosed in parentheses and brackets. For the next deliverable, we plan to eliminate parenthetical expressions that are separated from the rest of the text by hyphens and commas. Eliminating such expressions is a fairly safe approach because it does not detract from readablity and these asides do not usually contain necessary information. Furthermore, their removal does not affect the rest of the sentence structure, so the rest of the sentence remains structurally intact without them.

Along these lines, we also chose to remove adverbs, since they are generally extra descriptors that do not contain content information. Their removal does not affect sentence readability.

Lastly, we wanted to maximize the amount of content we included up to the 100 word limit. In doing so, we wanted to account for the case where we had a summary of 70 words, but the next best ranked sentence would put us over 100 words. We opted to fill in the summary with the next best ranked sentence that would bring the summary as close to 100 words as possible without going over the limit.

For the next deliverable, we would like to incorporate additional heuristics that will address some of the issues we found in our output summaries (see Qualitative Analysis section below for additional details). For example, we noticed many sentences of the following form:

> The bank at southwest China's Giant Panda Protection and Research Centre in the Wolong Nature Reserve in

	ROUGE-2 (Average F)
D2 System	0.05031
LEAD Baseline	0.05376
MEAD Baseline	0.05927

Table 1: Comparison of our D2 system to baselines. The average ROUGE-2 F-measure is reported.

> Sichuan province will be completed this year, the China Daily said.

This example, taken from D1003, shows a statement that is attributed to a particular source, but is not a quotation. The final subordinate clause could be removed and the information would be preserved. We plan to introduce a heuristic to handle this situation, since it appeared frequently in our summaries.

So far, the approaches we have discussed involve heuristics that deal with the surface level of the sentences. For the next deliverable, we want to incorporate some deep processing methods, and will explore approaches involving learning and phrase structure.

4 Results

A comparison of our system to baselines is shown in Table 1. There is no statistically significant difference between ROUGE-2 F-score achieved by our system and that of the LEAD and MEAD baselines. In addition to computing ROUGE scores, we also qualitatively evaluate our summaries.

4.1 Automatic Evaluation

We use a standard implementation of ROUGE to evaluate our summaries. The metric measures overlap between automatic and human summaries. Two types of ROUGE scores were used: ROUGE-1 measures unigram overlap, and ROUGE-2 measures bigram overlap. These scores have, in aggregate, been shown to correlate well with human assessments of responsiveness and to work well for extractive summaries (Rankel et al., 2011). A higher ROUGE score signals better overlap with the human summary.

The average ROUGE-1 F-score was **0.21316**, and the average ROUGE-2 F-score was **0.04340** (95% CI [0.04008-0.06069]). We compare on ROUGE-2 against two standard baselines. Our system performs nearly as well as the baselines, and we find no statistically significant difference. The LEAD baseline takes the first 100 words as the summary, while MEAD is an exemplar centroid-based summarization system (Radev et al., 2001).

4.2 Qualitative Analysis

Apart from evaluating with ROUGE, we randomly select and review summary outputs. We observe two main issues in our summaries, as shown in Figure 4. First, **attribution** to the source of the information adds unnecessary text that is irrelevant to the main topic. Secondly, we find in our summaries a great amount of **redundancy**. There are many repeated terms, and the summaries contain sentences that are essentially paraphrases of one another.

5 Discussion

Some of the results we saw the presence of attribution and this can be due to the fact that there is no filtering mechanism to remove these phrases in text. Appropriate filtering methods to identify and remove attribution can reduce the words in a summary, giving way for more important content. Another aspect we found was the presence of redundancy in the summaries produced. This can be solved by using sentence similarity, thus if the sentence similarity is high then the content between the sentences is redundant. We had tried using sentence similarity scores to eliminate redundancy, however, this information is not retained when the data is passed through the LDA module.

In the next iteration of the development process, we are aiming to experiment with hyper parameters in LDA and also augment LDA topic modeling with TF-IDF and concreteness scoring as well as delexicalization of the input based on word vector similarity. Another option is to explore skip-thought vector sentence representations. For information ordering, we plan to use the document meta-information and to experiment with chronological ordering and promote coherence using cosine similarity. For content realization, we aim to use phrase structure to ensure cohesion and experiment with learning methods.

6 Conclusion

An extractive multidocument summarization system was successfuly implemented using topic modeling with LDA as the main content selection method and surface level filtering of the sum-

D1002

Some of the police officers who shot and killed Amadou Diallo have told associates that their attention was drawn to Diallo when they saw him standing on the stoop of a Bronx apartment building and thought they saw him peering into the window of a first-floor apartment~~, according to people with knowledge of the case.~~

D1014

Many spoiled teenagers from one-child families eat a particular kind of food, such as meat or sweets, which might lead them to develop **obesity**, diabetes or heart disease.

The study shows that the **12 percent obesity** is to the standard in developed countries.

The research showed that the ratio of adults suffering from **obesity** has reached **12.6 percent**, and that of youngsters is 11.35 percent.

Figure 4: Summary examples from two topics. The summary for **D1002** on the left shows the problem of including extra attribution information. The summary for **D1014** on the right shows the problem of including redundant terms.

maries produced by the system. The end-to-end system achieved the ROUGE-1 and ROUGE-2 scores comparable with the baselines and outperformed the baseline in one out of three cases. The scores show that our approach has the potential to yield good results. Our first implementation was largely focused on working out the system architecture and ensuring that the separate components run smoothly following one another. We also experimented with several content selection methods and built the scaffolding for a more refined versions of this module.

References

Regina Barzilay, Noemie Elhadad, and Kathleen R. McKeown. 2002. Inferring strategies for sentence ordering in multidocument news summarization. *J. Artif. Int. Res.*, 17(1):35–55, August.

M. Brysbaert, A.B. Warriner, and V. Kuperman. 2014. Concreteness ratings for 40 thousand generally known english word lemmas. *Behavior Research Methods*, 46(1):904–911, August.

Dragomir R Radev, Sasha Blair-Goldensohn, and Zhu Zhang. 2001. Experiments in single and multidocument summarization using mead. In *First document understanding conference*, page 1À8. Citeseer.

Peter A Rankel, John Conroy, Eric Slud, and Dianne P O'leary. 2011. Ranking human and machine summarization systems. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 467–473.