NLP Systems and Applications: Automatic Summarization

Claude Zhang Julia McAnallen Genevieve Peaslee Zoe Winkworth
University of Washington, Seattle
{youyunzh, jmcanal, genevp, zoew2}@uw.edu

Abstract

This paper describes the design and implementation of three related multidocument summary generator systems: a baseline lead sentence system, a system modeled on MEAD (Radev et al., 2001); and MELDA, an expansion of MEAD that incorporates enhancements with LDA topic modeling (Blei et al., 2003). The two baseline systems show improved performance from the first report, as a result of modifications to MEAD weighting and several bug fixes. When run on the newswire document sets originally used in the 2010 TAC (Text Analytics Conference) summarization shared task LEAD sentence has a ROUGE-1 recall value of 0.18494, while MEAD ROUGE-1 recall value had a maximum of 0.19726. MELDA has peak ROUGE-1 scores of 0.16827, 0.21598, and 0.18818.

1 Introduction

We present a baseline multi-document summarization system with three different content selection strategies, for comparison: lead sentence, MEAD score-based (Radev et al., 2001; Radev et al., 2002; Radev et al., 2004), and MEAD with LDA topic modeling enhancements to content selection and information ordering. We compare ROUGE scores on output summaries produced by different parameter combinations and discuss the effect of these results on our plans for future work.

2 System Overview

Our systems follow the summarization steps outlined in Figure 1 and our modules are also structured based on this flow.

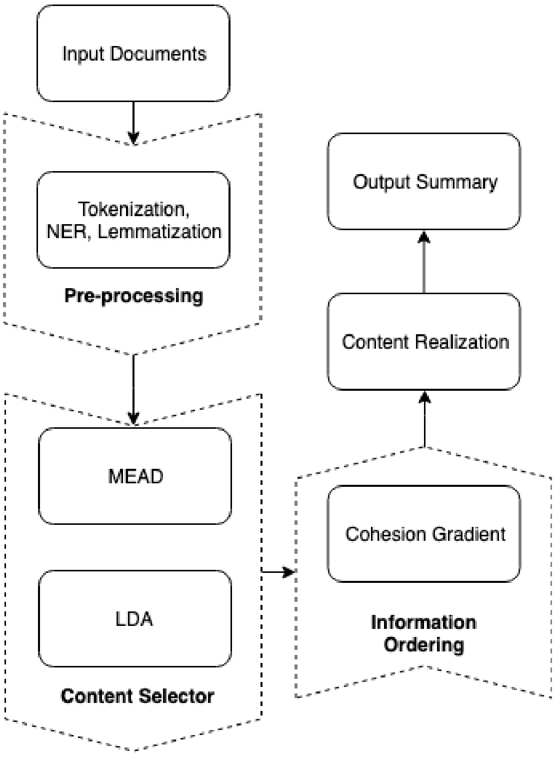

Figure 1: System Architecture Diagram

2.1 Pre-processing

We use nltk's tokenizer to segment documents into sentences. The segmentation this produces is not perfect but an attempt to identify and exclude output strings that are clearly not valid sentences, while helping readability, added too much runtime and did not help the ROUGE scores, so we keep the entire output of the segmentation process. We then use the spacy package to tokenize each sentence, as well as extract and link the named entities. Sentences that are all stopwords or all named entities are excluded from the content selection process.

In order to more accurately measure sentence similarity, the most recent iteration of the system bases the sentence vector representations on sets of tokens that reflect lemmatization and named entity recognition pre-processing steps. We attempted to also replace all co-referring spans of text with one unique mention per referent (helping sentences with similar meanings to look more similar on the surface) but this added a significant amount of processing time and actually gave lower ROUGE scores.

2.2 Summary Generator Module

The summary generator module of our system handles pre-processing and initiates the three core steps of the summarization task - content selection, information ordering, and content realization, described below - for each topic in the input.

2.3 Content Selector Module

The content_selector module selects sentences that are most salient to the topic from the set of topic documents, as evaluated by the given selection strategy. The three content selection strategies we implemented are represented by three different content_selector modules.

2.4 Information Ordering Module

In MELDA, the updated system, a new information ordering module was added, info_ordering. This module specifically addresses inter-sentence cohesion with a method we call a cohesion gradient.

3 Approach

For each of our content selection strategies, we implemented the three core subtasks of the summarization task as described above.

3.1 Lead Sentence System

Our lead sentence system generates a summary by selecting from only the first sentence in every document.

3.1.1 Content Selection

To select the content eligible to be included in the summary, we simply select the first sentence in each document included in the topic.

3.1.2 Information Ordering

Sentences selected by the content selector are ordered chronologically by document date and then by article ID.

3.1.3 Content Realization

The final output contains sentences as they appear in the original documents, without editing. Sentences are added to the summary one by one, most recent first, until the addition of the next sentence would make the total word count of the summary greater than 100. Only full sentences are added to the summary, but sentences that are over 100 words are skipped to prevent empty outputs.

3.2 MEAD-based System

Our MEAD system selects content by choosing sentences with the highest MEAD scores. The MEAD score is composed of the sentence position score, the first sentence overlap, the centroid score and a redundancy penalty.

3.2.1 Content Selection

The MEAD score calculation has four components. A centroid score, a positional score, and a first sentence-overlap score are optionally weighted and summed to get a preliminary score for each sentence. The fourth component is a redundancy penalty applied to all remaining sentences each time an individual sentence is added to the summary.

Centroid Score A centroid vector is computed for each topic by multiplying count and IDF, following Radev (2004). Count values are the average counts of words in a given topic, which is then multiplied by IDF values calculated from an external corpus. Both the Reuters and Brown Corpora from NLTK were tested as external corpora (Bird et al., 2009). Brown resulted in marginally better final results, and was chosen as the base corpora for the MEAD submission.

Next, a predefined threshold value is applied to the cluster centroid; words with centroid values below the threshold are set to zero. We developed threshold settings based on the top quartile, mean, and bottom quartile of word centroid values. (Radev (2004) did not provide guidance on selecting a threshold value.)

A centroid score for each sentence is then calculated by summing the centroid value for each word in the sentence after the threshold has been applied:

$$C_s = \sum_{w} C_{w,s}$$

where C_s is the centroid score of sentence s, w

ranges over all the words in the sentence, and $C_{w,s}$ is the centroid value for each word w in sentence s.

Positional Score The positional score P_s for each sentence is the sentence's position in the document (s; 1st = 0, 2nd = 1, etc.) scaled by the distance from the beginning of the document:

$$P_s = \frac{n - s}{n}$$

where P_s is the positional score of sentence s and n is the number of sentences in the document. Note that our calculation diverges from the equation provided in Radev (2004) in two ways. First, they add one to the numerator under the assumption that the first sentence has a position score of 1; however, we used Python file IO functions that start numbering at 0. Second, the positional score is often scaled against a maximum centroid score, which we did not use for this baseline.

First Sentence Overlap Score Overlap with first sentence is calculated using cosine similarity to compare each sentence with the first sentence in the document containing it.

$$F_s = \frac{S_0 \cdot S_s}{||S_0|| \times ||S_s||}$$

where F_s is the first sentence overlap score of sentence s and S_s is the TF*IDF-weighted vector representation of sentence s. Note that this differs from the implementation of Radev (2001) who use the inner product of the TF*IDF-weighted vector representations of a given sentence and the first sentence.

MEAD Score The scores for centroid, position and first sentence overlap are summed for all sentences in the document.

$$score = w_c C_s + w_p P_s + w_f F_s$$

where w_c, w_p, and w_f are optional weights applied to the scores. Note that we also depart from (2001) here since they normalize all three features in the range 0 - 1 while we leave them as is.

Redundancy Penalty The redundancy penalty is calculated each time a sentence is added to the summary to prevent redundant sentences from being included. Each sentence is compared to the last sentence added to the summary, and the penalty is computed by dividing the number of overlapping tokens by the number of tokens in the sentence pair and doubling the result. This penalty is then subtracted from all the sentence scores.

$$R_s = 2 \times \frac{S_s * S_l}{cnt(S_s + S_l)}$$

where R_s is the redundancy penalty for sentence s and S_s is the TF*IDF-weighted vector representation of sentence s and S_l is the TF*IDF-weighted vector representation of the sentence most recently added to the summary.

3.2.2 Information Ordering

Sentences selected by the content selector are ordered by descending MEAD score. After each sentence is added to the summary, scores are recalculated by subtracting the redundancy penalty and all sentences are re-ordered by the new scores before the next summary sentence is added.

3.2.3 Content Realization

The final output contains sentences as they appear in the original documents without editing. As in the lead sentence implementation, sentences are added to the summary one by one until the addition of the next highest-scoring sentence would push the total word count of the summary over 100 words. Only full sentences are added to the summary, but sentences that are over 100 words are ignored, to prevent empty summaries.

3.3 MELDA System

3.3.1 Content Selection

The foundation of MELDA content selection is the approach described for MEAD in §3.2.1. MELDA additionally uses LDA topic modeling (Blei et al., 2003) to choose the top sentences according to both their MEAD scores and their LDA topic scores. Therefore, top MEAD scores that are also highly representative of a topic are more likely to be selected than high scores that are not highly representative of a topic.

Latent Dirichlet Allocation The LDA method is used to build statistical models that classify text in a document into abstract "topics". To avoid redundancy, we will use the term "topics" for the LDA extracted topics and "cluster" for the document themes.

In this system we use the gensim package to build an LDA model over the text of all the documents in a cluster and get the LDA topic scores

	R1-R	R1-P	R1-F1	R2-R	R2-P	R2-F1
Lead	0.18494	0.23168	0.20364	**0.04753**	**0.06031**	**0.05247**
MEAD	**0.19726**	**0.23328**	**0.21247**	0.04378	0.05176	0.04717
MELDA-FILTERED	0.10885	0.17431	0.13161	0.01853	0.03020	0.02245
MELDA-COREF	0.16529	0.19621	0.17822	0.03607	0.04200	0.03851
MELDA-LDA-NORM	0.10379	0.16613	0.12608	0.01809	0.02929	0.02204
MELDA-NORMS	0.10344	0.16665	0.12603	0.01731	0.02826	0.02119
MELDA	**0.16827**	**0.21598**	**0.18818**	**0.03968**	**0.05182**	**0.04474**
MELDA-L	0.11380	0.16768	0.13357	0.01920	0.03096	0.02334
MELDA-M+	0.16808	0.21711	0.18783	0.03943	0.05149	0.04432
MELDA-M-	0.16956	0.21707	0.18867	0.04034	0.05151	0.04490
MELDA-C-	0.16307	0.20500	0.18022	0.03715	0.04668	0.04112
MELDA-25	0.17274	0.21378	0.18980	0.03741	0.04725	0.04149
MELDA-52	0.17091	0.21749	0.18952	0.0360	0.04608	0.04008
MELDA-33	0.16890	0.21216	0.18662	0.03774	0.04799	0.04197
MELDA-55	0.15842	0.21083	0.17920	0.03320	0.04442	0.03762

Table 1: ROUGE Results MELDA

for each sentence, which represent the probability that a sentence belongs to each of the abstract topics. Considering that the length of the summary is 100 words and the average sentence length is around 20, for now we set the number of topics to three by default. We then tried different settings for other parameters; the results are in §4.

To integrate the LDA and MEAD score, we simply add the MEAD score for a sentence to each LDA topic score to compute MELDA scores and select content based on the result.

3.3.2 Information Ordering

Information ordering in MELDA is more sophisticated than the MEAD approach of ordering the top scoring sentences by score. We instead leveraged LDA to improve coherence between sentences.

Cohesion Gradient The LDA scores calculated for content selection were used to improve information ordering in the system. In particular LDA topic scores for the top sentences were used to order the sentences according to a cohesion gradient, which aimed to improve summary coherence.

Before applying the cohesion gradient, the content selection process pre-selects a fixed number of sentences to be ordered. Then, the cohesion gradient function is applied as follows: first, the most prevalent topic among the lead sentences across the entire document cluster is chosen. Then the sentence in the pre-selected set with the highest topic score for that topic is chosen as the first sentence of the summary. The sentence with the next highest topic score for the topic is chosen next, and this process is repeated until another topic takes over as the highest valued topic for a sentence. Each time this happens, the topic switches.

In this way, the information in the summary is ordered so that sentences more gradually transition between topics, hence are more cohesive. In reality, with short summaries that are rarely more than three sentences, we expect a maximum of two topic shifts per summary.

The goal is to order information first by most salient topic overall among lead sentences, then gradually shift topics from sentence to sentence to avoid jarring topic changes, as well as "flip-flopping" back and forth between the same topics.

3.3.3 Content Realization

In the current implementation, as in MEAD, sentences appear in their original form.

4 Results

The baseline results from LEAD and MEAD are reported in the first section of Table 1. Only the top scoring MEAD configuration is listed.

ROUGE results for various configurations of MELDA are in the second section, starting with row 3. For all MELDA configurations, the underlying MEAD parameters, unless specified otherwise, are: Brown corpus (NLTK version) for IDF calculations, MEAD score weights of 1-1-1, three LDA topics, five sentences per topic, and no normalization for sentence length. The baseline MELDA score with this configuration is in the third row; this is followed by scores for MELDA filtered with more strict sentence pre-processing to exclude sentences with too much punctuation or too many named entities; MELDA with coreference resolution, using the Hugging Face neural coreference package; MELDA with LDA scores normalized for sentence length; MELDA with both LDA and centroid scores normalized for sentence length.

In the next section of Table 1, results are reported for different configurations of the underlying MEAD centroid parameters. MELDA-C- sets the MEAD centroid score to zero, effectively replacing the centroid score with the LDA topic score; MELDA-L sets all the MEAD scores to 0, thus weighting only according to LDA topics. MELDA-M+ weights the MEAD score higher than the LDA topics, whereas MELDA-M- weights the MEAD score less than LDA topics.

In the last section of Table 1, scores are reported for different hyper-parameters for the number of LDA topics and the number of sentences per LDA topic. The first number is the number of topics and the second number is the number of sentences, e.g. for MELDA-25, there are 2 topics and 5 sentences per topic.

For comparison, below are two example summary outputs from the document cluster with ID D1029.

MELDA summary:

> *Thousands of government workers and their families have evacuated a remote town in southwestern Pakistan, a senior official said, amid fears of renewed fighting between renegade tribesmen and security forces after clashes left at least 30 people dead.*
> *Fierce gun battles between tribal rebels and Pakistani troops in the troubled southwestern province of Baluchistan have left up to 30 people dead and more than 70 injured, an official said Friday.*

MEAD summary:

> *Pakistan has deployed police commandos to guard trains and railway stations after frequent blasts in its insurgency-hit southwestern province, officials said Thursday.*
> *A bomb exploded near a railway track in Pakistan's southwestern Baluchistan province on Friday, injuring two soldiers, an official said.*
> *Tribal rebels attacked a paramilitary convoy in restive southwest Pakistan on Thursday, killing three soldiers and injuring seven others, a senior military officer said.*
> *Four people were injured and a powerline and a government office damaged in three separate bomb blasts in southwestern Pakistan's restive Baluchistan province, officials said Sunday.*

5 Discussion

Bug fixes and a score normalization step added in this latest iteration boosted ROUGE scores for the LEAD and MEAD systems. The scores for the top MELDA configuration are lower than these two baseline systems however, indicating room for future work.

The work on coreference resolution that went unused in content selection lays the groundwork for improvements in content realization, as the basis for correctly handling references. We also plan to explore other methods for content realization such as removing attributive and gerund clauses, sentence-initial adverbs, and bylines and other artifacts of news article structure that are irrelevant to the summarization process.

6 Conclusion

We developed two baseline systems: a lead-sentence system and a system based on the original MEAD model (Radev et al., 2001), as well as an expanded system called MELDA, which adds LDA topic modeling to the MEAD score calculation process. LDA is used to weight content selection by choosing sentences that represent the most salient topics accordng to the LDA topic groupings. Then, in information ordering, topic scores are used to perform a cohesion gradient ordering process to improve inter-sentence cohesion of the summaries. The outputs of the MELDA system are overall comparable to MEAD with slightly lower ROUGE scores.

References

Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural Language Processing with Python*. O'Reilly Media.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

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.

Dragomir Radev, Adam Winkel, and Michael Topper. 2002. Multi document centroid-based text summarization. In *ACL 2002*. Citeseer.

Dragomir R Radev, Hongyan Jing, Małgorzata Styś, and Daniel Tam. 2004. Centroid-based summarization of multiple documents. *Information Processing & Management*, 40(6):919–938.