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 two baseline multidocument summary generator systems: a lead sentence system and a system modeled on MEAD (Radev et al., 2001). The two systems performed comparably when run on the same newswire document sets (those originally used in the 2010 TAC (Text Analytics Conference) summarization shared task): Lead sentence has a ROUGE-1 recall value of 0.13550, while MEAD ROUGE-1 recall values range from 0.11392 to 0.16157 depending on hyperparameter settings. The MEAD system serves as an adequate baseline for planned upgrades, including the incorporation of LDA topic modeling to improve content selection.

1 Introduction

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

2.1 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.

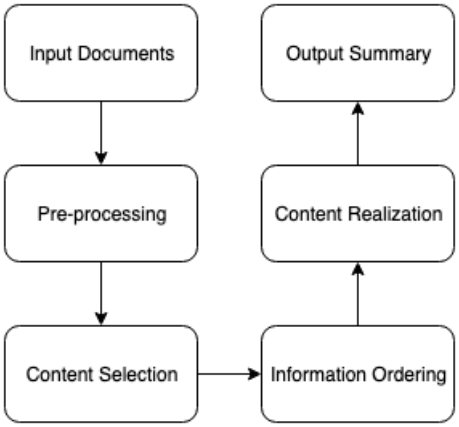

Figure 1: System Architecture Diagram

2.2 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 two content selection strategies we implemented are represented by two different content_selector modules.

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*IDF, following Radev (2004). Count values are the average counts of words in a given topic cluster, 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 provide 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.

4 Results

We ran our MEAD-based system with a variety of parameters. ROUGE results for each combination are listed in Table 1. The first two rows, MEAD-R and MEAD-B, show the difference between using the Reuters and Brown corpora in the Count*IDF calculation; weighting is equal between all three preliminary score components and the threshold for the centroid vector is is the mean value across word centroid values.

The next section of the table compares different thresholds a word's Count*IDF score must exceed to be included in the centroid vector. MAX is the top quartile of centroid values (the average of the maximum centroid value and the mean of all the centroid values); MIN is the bottom quartile of centroid values (the average of the minimum centroid value and the mean of all centroid values); and 0 is no threshold.

The remaining rows show different weight ratios tested. The first number represents the weight applied to the centroid score (w_c), the second, to the position score (w_p), and the third to the first sentence overlap score (w_f).

For comparison, below are two example summary outputs from the document cluster with ID D1046.

Lead summary:

> *In a report prepared for the meeting in Jakarta, the World Bank said recent economic reforms by Jakarta have "boosted Indonesia's economic resilience and positioned it better to absorb massive shocks, such as the natural disaster in Aceh and North Sumatra."*
> *Indonesia's death toll stands at almost 115,000, with thousands more still missing and hundreds of thousands left homeless after entire towns and villages were swept away or reduced to rubble by the quake and waves.*

MEAD summary:

> *Uzbekistan on Monday sent a plane with 35 metric tons (39 short tons) of humanitarian aid to tsunami-hit Indonesia, the Foreign Ministry said.*
> *The Ilyushin-76 cargo plane headed for Medan, the main city on Indonesia's Sumatra island, one of the areas hardest hit by the Dec. 26 tsunami, the ministry said in a statement.*
> *A second plane with an additional 16 metric tons (18 short tons) of aid was to depart Monday evening. The aid includes medicines, tents, food, two off-road vehicles, a motor boat, field kitchens, drinking water and helicopter parts.*

5 Discussion

The results for the MEAD summary generator system are only marginally higher than the

	R1-R	R1-P	R1-F1	R2-R	R2-P	R2-F1
Lead	0.13550	0.14145	0.13792	0.02377	0.02469	0.02417
MEAD-R	0.10262	0.14015	0.11646	0.02044	0.02753	0.02308
MEAD-B	**0.11718**	**0.15493**	**0.13160**	**0.02518**	**0.03231**	**0.02806**
MEAD-MAX	**0.13316**	**0.16314**	**0.14540**	**0.02818**	**0.03445**	**0.03078**
MEAD-MIN	0.11392	0.15110	0.12834	0.02509	0.03226	0.02799
MEAD-0	0.11490	0.15216	0.12909	0.02500	0.03200	0.02782
MEAD-1-0-0	0.12760	0.15659	0.13947	0.02680	0.03235	0.02910
MEAD-0-1-0	0.12369	0.15883	0.13787	0.02339	0.03028	0.02619
MEAD-0-0-1	0.12566	0.15788	0.13905	0.02613	0.03368	0.02924
MEAD-0-1-1	0.12964	0.13895	0.13320	0.02244	0.02353	0.02291
MEAD-1-0-1	0.13170	0.16117	0.14368	0.02754	0.03348	0.02999
MEAD-1-1-0	0.13290	0.16290	0.14514	0.02820	0.03444	0.03079
MEAD-1-10-10	0.15786	0.19636	0.17323	0.03533	0.04577	0.03944
MEAD-1-10-5	0.16157	0.19089	0.17330	0.03908	0.04662	0.04212
MEAD-1-10-1	0.14543	0.17971	0.15904	0.03170	0.03942	0.03481
MEAD-1-20-5	0.15762	0.19021	0.17056	0.03742	0.04646	0.04094
MEAD-.5-1-.5	0.13668	0.16838	0.14912	0.02909	0.03511	0.03154

Table 1: ROUGE Results

much simpler Lead sentence summary system. Lead sentence has a ROUGE-1 recall value of 0.13550; MEAD ROUGE-1 recall values range from 0.11392 to 0.16157. While we had hoped for greater improvement, we believe that the MEAD system serves as an adequate baseline for the next steps of the summarization project.

Our results so far point to some specific next steps for improvement. We currently rely on NLTK's tokenizer to separate the raw document text into sentences, but the training data for this system does not lend itself to that tool's approach. For example, we would like to incorporate our own tests for sentence boundaries. An example of an undesirable sentence returned by the NLTK sentence tokenizer is shown below:

> *HONG KONG, January 23 (Xinhua) – Following are Hang Seng Index (key indicator of Hong Kong's blue chips), Hang Seng China Enterprises Index and the turnover on the Hong Kong Stock Exchange today (Thursday):*
> *Index: Thursday Wednesday Change*
> *Hang Seng Index (HSI) 13,610 13,692 -82*
> *HSI Sectors:*
> *Finance 14,545 14,534 +11*
> *Turnover (Million HK Dollars):*
> *HSI 8,285 11,158 -2,873*
> *(Million US Dollars) 1,062 1,439 -377*

While this is an appropriate sentence from the NLTK sentence tokenizer's perspective, it is not a useful sentential unit for our task. This sentence also received a high MEAD centroid score, which is likely due to a combination of its vocabulary and a high level of overlap with other sentences in the topic cluster. However, from a coherence and readability perspective, it is a poor choice to include in a summary. Even some simple pre- or post-processing measures, such as ignoring "sentences" containing multiple newlines, would be helpful in preventing sequences like this from appearing in output summaries.

We also intend to incorporate additional steps in the pre-processing stage, including named entity recognition, lemmatization, and coreference resolution, which we expect to help in identifying salient sentences. In general, we feel that a more sophisticated strategy for choosing the sentences returned by the content selector will improve results.

The most substantive change we propose in the next stage is to introduce topic modeling (specifically, LDA) into the scoring of sentences during content selection. Also, as part of the content selection step, we will explore using other external corpora for the IDF calculation.

6 Conclusion

We developed two baseline systems: a lead-sentence system and a system based on the original MEAD model (Radev et al., 2001). While the ROUGE scores for the MEAD system do not indicate high performance, we believe it serves as an appropriate baseline. Since we developed the system in a modular fashion, with highly compartmentalized functions, going forward we can easily modify components of the system to test a variety of system upgrades in the different system sub-components.

References

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

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.