Content Selection for Multidocument summarization - combining graph based ranking and topic modeling

Krishna Choksi
CIS
University of Pennsylvania
krishna@seas

Madhura Raju
CIS
University of Pennsylvania
rmadhura@seas

Varsha Shankar
CIS
University of Pennsylvania
vasha@seas

Abstract

In this work, we present three methods for performing content selection for extractive multi-document summarization. We implemented and analyzed three systems. The first system performs page ranking on sentences represented as a graph with edge weights between vertices being the number of shared topic words. The second system performs topic modeling on the source documents and ranks sentences using cross linked entities and topic proportions of the sentences using the PageRank algorithm to get a topic-rich set of sentences. The third method combines the aforementioned two approaches by utilizing word similarity metric and KL-divergence. We compare each system with a baseline summary generated based on sentence centrality and we report ROUGE scores on all of them.

1 Project Overview

Content selection is one of the major areas of research in the field of multi-document summarization. For this project, we implemented two approaches, the first performs graph based ranking on sentences and the second, we extract named entities from each sentence and for sentences which share entities, use their topic proportion similarity to form a matrix which is fed into the page rank algorithm to rank the sentences. We finally combine the summaries generated by these two approaches by a combination of KL Divergence and Cosine Similarity to sample the content for a combined summary. We test these three systems against the baseline system which ranks sentences based on centrality

2 Graph based sentence ranking and selection

2.1 Overview

Graph based ranking is most popular in applications involving the Internet, most famously so by Google for indexing web pages using the PageRank (Brin and Page, 1998) algorithm. A precursor to this algorithm for page ranking was HITS (Kleinberg, 1999). These proven and successful ranking algorithm can be applied to summarization tasks when documents are represented in the form of a structure with nodes and edges resembling a graph.

In our implementation, we have modeled an entire document collection as a graph with each sentence representing a node. Nodes are connected by edges if they share atleast one topic word in common. The strength of the edges are calculated as a proportion of the sum of the chi-square scores of the common topic words to the sum of the logarithmic lengths of each sentence. Using PageRank formulations for undirected graphs to include the weights on the edges, we ranked sentences and extracted the most highly ranked nodes or sentences.

Even with very basic edge linking policy and weighting, we have seen vastly improved summaries

that are very readable and coherent.

$$Weight(S_i, S_j) = \frac{\sum \chi^2(W_k) | W_k \in S_i, W_k \in S_j}{log(|S_i| + log|S_j|)}$$

2.2 Related Work - Text Rank

Sentences in the document were treated as vertices or nodes of a graph. An edge between two nodes indicated a relation between the two sentences. The number of shared words between two sentences served as the weight on the edge joining their respective vertices in the graph. (Mihalcea, 2004,)

$$Similarity(S_i, S_j) = \frac{W_k | W_k \in S_i, W_k \in S_j|}{log(|S_i| + log|S_j|)}$$

Using these edge weights, graph ranking algorithms were run, to generate ranks for the nodes/vertices of the graph. Sentences were then sorted in reversed order of their score, and the top ranked sentences were selected for inclusion in the summary.

Their technique was evaluated for all three graph ranking algorithms (HITS, PositionalPower, PageRank) on undirected, directed forward and directed backward graphs in the context of a single document summarization task.

TextRank uses a non-local approach to sentence selection (i.e. sentences were not scored individually, rather on the basis of similarity ratings to other sentences) which was appealing. Unlike the similarity rankings that we built for sentences during the homework where we made a single pass over the text to make one-to-one cumulative similarity comparisons between sentences and picked the sentences that were most similar to the rest of the document, TextRank employs a converging random walk algorithm to decide the relevance of each sentence starting with an initial similarity metric value.

However, we do not entirely agree with the initial similarity metric that was used (count of common words between sentences to decide an edge weight).

In a preliminary implementation that uses PageRank on an undirected graph of the documents in a collection, we found that using even a common topic word count between two sentences instead of just common words gives a nicer selection of sentences.

2.3 Related Work - TextRank With Shortest Path

The approach used was similar to the above TextRank save for the selection of sentences, that is, the sentences were scored using the number of shared words between two sentences. However, sentence selection was performed by finding the shortest path between the first node (first sentence in the document) and the last node (last sentence in the document) using priority queueing based on the edge weights. (Thakkar, Dharaskar and Chandak, 2010,)

Intuitively, it seems like a good way to produce better flowing summaries. The evaluation on the technique however, is not presented in quantitative terms. It might have made a more convincing argument if they had created an entity grid based evaluation of the summary coherence and salience. Even a manual evaluation of the summaries would have sufficed. Instead they have just claimed to produce smoother summaries although smoothness is hard to quantify objectively.

3 Ranking using Cross-linked Entities and Topic Proportions of Sentences

3.1 Overview

We use the generative probabilistic model, Latent Dirichlet Allocation(We use an available implementation from the internet), to represent a collection of documents in terms of topic probabilities. Every element in the collection is modeled as an infinite mixture over an underlying set of topic probabilities. We have utilized an available implementation of the algorithm and modified it to create the vector representations at all four levels, viz. words, sentences, documents, and collection of documents or corpus. For our implementation, we choose the number of topics to be 5.

Now, we have every sentence in the corpus and their corresponding topic proportions. We take every sentence in the corpus and extract all the Named Entities in each of them and find sentences which have cross-liked entities i.e. we are now able to tell if two sentences mention the same named entity or not.

We then form the square matrix M_{NXN} where, $M(i,j) = 0$ if sentence i and j do not have any cross linked entities or if i == j and $M(i,j) = Sim(T_i, T_j)$ where T_i and T_j are the topic proportions of the sentences i and j (learned from LDA) and Sim is a vector similarity function.

For Sim, we use the extended jaccard metric which computes vector similarity as follows.

$$ExtendedJaccard(I,J) = \frac{\sum_{i \in I, j \in J} min(i,j)}{\sum_{i \in I, j \in J} max(i,j)}$$

We then normalize each row of the matrix and feed it as the input to the PageRank algorithm and sort the sentences by their decreasing order of the pageRank value. We select the top sentences from this final list.

4 Combining the two Summaries

4.1 Overview

In this module, we made use of Cosine similarity and KL divergence. As we are working with multi-document summaries, it was not possible to order the sentences just by comparing with the actual articles. So, this was the most important thing to focus on while generating the final summary. The first sentence of the final output is chosen from either of the summaries based on KL values. KL divergence is used to calculate the similarity between two inputs. It is asymmetric i.e. KL(sentence1,sentence2) is not same as KL(sentence2,sentence1). So, we calculated the both of them. The value of KL specifies how much the 2nd sentence differs from the 1st sentence in reference to the words of the 1st sentence. So, if KL(sentence1,sentence2) is lesser than KL(sentence2,sentence1) ,where sentence1 is the first sentence of one summary and sentence2 is the first sentence of the other summary, then it means that sentence2 is the concise form of the both. So, on this basis we choose sentence1 as the first sentence of our final summary. This method will help us choose the sentence which conveys most information of the both while having less number of words.

Now, this chosen sentence becomes the reference for picking up the next sentence. We need to take into account three cosine similarities for it.

(1) between reference sentence and next sentence to be considered from the first summary

(2) between reference sentence and next sentence to be considered from the second summary

(3) between the two sentences to be considered from both the summaries.

If cosine similarities 1 and 2 are equal and cosine similarity 3 is greater than 0.3 then only of the sentences from the two summaries is chosen for the final summary and the sentence from the other summary is ignored. This is because cosine similarity 3 indicates that both sentences are highly similar and it would be fine to ignore one of the two sentences. But if cosine similarity 3 is lesser than 0.3 then both the sentences from the summaries should be included in the final summary as cosine similarity 3 indicates they are highly dissimilar and provide different information.

If cosine similarities 1 and 2 are different then the sentence from higher cosine similarity summary is included in the final summary. But the sentence from the other summary is not ignored. It will be taken into consideration later on. So whenever a sentence from a summary is appended to the final summary, that sentence becomes the reference sentence and the pointer is shifted to the next sentence in that summary. This process goes on until all the sentences from both the summaries are parsed.

The final step is to remove the very highly similar sentences of both the summaries. Each sentence of first summary is compared with all the sentences of the second summary and the highest cosine similarity is found. If the highest cosine similarity for a

sentence exceeds 0.3 then one of the two sentences is pruned from the final summary. Thus, final summary might be of lesser number of sentences compared to the sum of number of sentences of both the summaries.

5 Results

We evaluated the three approaches and the baseline by using ROUGE-1, ROUGE-2 and ROUGE-SU scores and we used four document collections from the DUC2004 corpus and we summarize the results in the following table.

Table 1: Average ROUGE Recall

Method	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.32089	0.05759	0.09207
Graph Based	0.35015	0.08558	0.11921
LDA Based	**0.55189**	**0.14964**	**0.27965**
Combined	0.51026	0.12120	0.23014

Table 2: Average ROUGE Precision

Method	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.14654	0.02606	0.01872
Graph Based	0.17095	0.04165	0.02832
LDA Based	0.16236	**0.04287**	0.02455
Combined	**0.18209**	0.04257	**0.02988**

Table 3: Average ROUGE F-score

Method	ROUGE-1	ROUGE-2	ROUGE-SU
Baseline	0.20048	0.03573	0.03082
Graph Based	0.22953	0.05597	0.04565
LDA Based	0.24973	**0.06632**	0.04480
Combined	**0.26796**	0.06291	**0.05273**

6 Conclusion

We found that the LDA based method outperformed all other systems in terms of ROUGE scores. Our hypothesis as to why this works best is as follows. LDA does a good job of finding clusters of words which are highly correlated and this framework enables us to model the percentage of topics which are covered in any chunk of text effectively. Sentences which have high similarity in their topic vectors tend to talk about the same topics. Hence, we have a dense graph of sentences where each sentence is linked to another sentence if they share entities and their link-strength is represented by the similarity of their topic proportions. PageRank is a fairly proven method for performing link analysis in such graphs to find those nodes(sentences in our case) which are important in the whole graph. Since, the link strength is related to content richness in our method, the higher the page rank, the more rich the content of the sentence.
Our hypothesis to why the combined approach did not work is that while combining the summaries our focus was on the ordering of the sentences for multi-document summaries and the ROUGE scores do not take into account the sentence ordering. May be some better evaluation approach can help justify this.

7 Future work

We didn't take into consideration any syntactic features and learn what type of sentence structure good summaries are made of. Additionally, we could also learn a ranker with these features and others like POS-tags ngrams or presence of different types of dependency relations etc and after we take a list of the most topic rich sentences, we could rerank them by this method to test whether it produced a more content rich summary.

We could also use coherence models and entity-grids to reorder the selected sentences to create a more coherent summary. Also, for ordering, we could learn a metric using the same set of features for content selection as to how the top part, middle part and the bottom part of the summary should consist of in terms of content. i.e. More pronoun constructs may appear in the middle and the bottom part whereas a very low number may appear at the top part of the summary.

References

R. Mihalcea. 2004
Graph-based ranking algorithms for sentence extraction, applied to text summarization

K. Thakkar, R. V. Dharaskar and M. B. Chandak. 2010
Graph-Based Algorithms for Text Summarization

David Blei, Andrew Ng and Michael Jordan. 2003
Latent Dirichlet Allocation

Larry Page, Sergey Brin and Rajeev Motwani. 1999
The PageRank Algorithm

KL Divergence code
http://staff.science.uva.nl/ tsagias/?p=185