Extractive Text Summarization using Latent Semantic Analysis
Natural Language Processing (Fall 2014) - Project

Aishwarya Padmakumar (CS11B004) and Dhivya Eswaran (CS11B012)

Indian Institute of Technology, Madras

Abstract. The availability of vast quantities of textual information today, and the inability of humans to summarize such quantities or process an excessive amount of information have contributed to the rising importance of text sumamrization. Unsupervised approaches to solve this problem are preferred due to data scarcity issues. In this project, we explore variants of the use of Latent Semantic Analysis for text sumamrization and attempt to incorporate diversity awareness in the same.

1 Introduction

With the explosion of data available on the Web in the form of unstructured text, efficient methods of summarizing text are becoming more important today. Text Summarization is the process of condensing source text into a shorter version, preserving its information content and overall meaning. Due to the inability of the human to assimilate vast quantities of information, crisp and relevant summaries are desirable. The key challenge in text summarization is inability to easily distinguish the more informative parts of a document from the less informative ones. Some classic use cases of text summarization are assimilation of news from multiple sources, analysis of blog or product review sites and summarization of knowledge sources like Wikipedia, journal articles and other publications.

The approaches to text summarization can broadly be classified into two categories - extractive and abstractive. Extractive techniques are more concerned with the identification of the sentences which are the most relevant to the idea conveyed by the entire passage. These sentences are then concatenated in the order in which they occur in the input text and presented to the user. Extractive text summarization presents multiple disadvantages: (1) the preferential selection of long sentences; (2) important information is usually spread across sentences, not all of which can be usually selected; (3) conflicting information may not be presented accurately; (4) the presented summary usually lacks continuity and problems in overall coherence are not uncommon. Due to these limitations, extractive summarization is typically followed by more sophisticated techniques that make the text cohesive.

Text summarization techniques may be classified into two types - supervised or unsupervised, depending on whether or not they rely on the availability of gold standard or ideal summaries for a set of training documents. Typically, it is extremely difficult to manually obtain model summaries due to disagreement between human authors about what constitutes the key points of a document. Hence, unsupervised methods are naturally more preferable.

Our project aims to explore variations of the Latent Semantic Analysis method to perform unsupervised extractive text summarization. The contributions of this project are two-fold: (1) We have compared and evaluated the performance using different term-sentence vector spaces (2) We attempt to incorporate diversity awareness into LSA by taking inspiration from Maximal Marginal Relevance (MMR), a measure prevalently used in the domain of Information Retrieval. We have tested the proposed methodologies on representative subsets three types of datasets - news, blog data and research papers.

2 Literature Survey

Text summarized is a fairly well-studied problem in literature right from the 1960s. One of the first attempts to solve this problem came from Luhn et al [15]. They used high-frequency words present in the document to score a sentence. From these early attempts to recent approaches such as those of Kedzie et al [12] that attempt to perform temporal summarization, text summarization has come a long way and a number of techniques have been proposed to solve the problem.

Summarization is broadly divided into two types: (1) extractive (2) abstractive. Extractive summarization identifies key texts based on statistical analysis of a number of features. In contrast, abstractive summarization focuses more on the expression of the main concepts in a document in clear language. [9] Extractive summarization techniques are broadly divided into two categories: (1) Supervised (2) Unsupervised. Supervised techniques such as those of Neto et al [18] and Kupiec et al [13] models text summarization as a classification problem and use a number of features to classify sentences as summary or non-summary sentences. Some examples of these include content and title words, sentence location, sentence length, upper case words and cue phrases. Different classifiers have also been used ranging from Hidden Markov Models (HMMs) [17] to Neural Networks [11]

Unsupervised approaches are preferred because, once their efficacy has been proved, they are applied to summarization in different domains even in the absence of model summaries in those domains. In Herandez et al [8], a sentence level bag-of-words model is built, weighted by TF.IDF and summary sentences are selected by similarity to a suitably defined query. This allows for the personalization of the summary by modifying the query used for extraction. This class of methods is referred to as topic-driven summarization, a paradigm of growing interest. Another work that makes use of the same model is that of Bookstein et al [2] that attempts to use clustering of TF.IDF vectors and selecting representative sentences from each cluster to form a summary. This method works on the assumption that sentences naturally cluster into the themes that the document addresses.

Latent Semantic Analysis was first proposed by Deerwester et al [6] as method for automatic indexing and retrieval in order to improve the detection of relevant documents. A basic use of LSA for text summarization is to extract the sentence that best explains each identified latent concept, as demonstrated by Steinberger et al. [20]. More interpretable concepts can also be used as dimensions. An example is the work of Wang et al [21] that makes use of concepts from HowNet. Our work extends on the work of [20] incorporating a term to account for diversity in the sentences selected.

Another approach that has been used is to exploit the presence of lexical chains, originally proposed by Barzilay et al [1] and computed efficiently by Silber et al [19].

Graph theoretic approaches are also popular. Works such as those of Kruengkari et al [**?**] model a document as a graph where sentences are nodes and two nodes are connected by a link if the sentences are sufficiently similar. Partitions in the graph correspond to topics and representative sentences can be selected using graph properties.

Systems have also been designed to perform multi-document summarization, such as the NeATS system [14] that performs extractive content generation, followed by the use of linguistic techniques and Maximal Marginal Relevance for filtering, and the BAYESUM system [5] that uses query focussed summarization techniques. Multi-lingual summarization is also an interesting area of research. Some breakhroughs in this regard are SimFinderML [7] and the MINDS system [4], both of which use an ensemble of standard techniques to perform robust summarization.

3 Methodology

3.1 The LSA method

The basic intuition behind the use of LSA in text summarization is that words that usually occur in related contexts are also related in the same singular space. LSA transforms sentence vectors from a term-space of non-orthogonal features to a concept-space of lower dimensionality with an orthogonal basis. This is done by performing singular value decomposition of term sentence matrix A. This gives us three matrices - U, V and Σ such that

$$A = U\Sigma V^T \tag{1}$$

where U and V are orthogonal matrices and Σ is a diagonal matrix whose diagonal elements represent the relative importance of each concept dimension in the basis of the concept space. The columns of U and the rows of V^T are respectively the vectors corresponding to the terms and sentences in the concept space. By retaining on top k concepts (in terms of their eigen values), we the best k-rank approximation to A in the least squares sense.

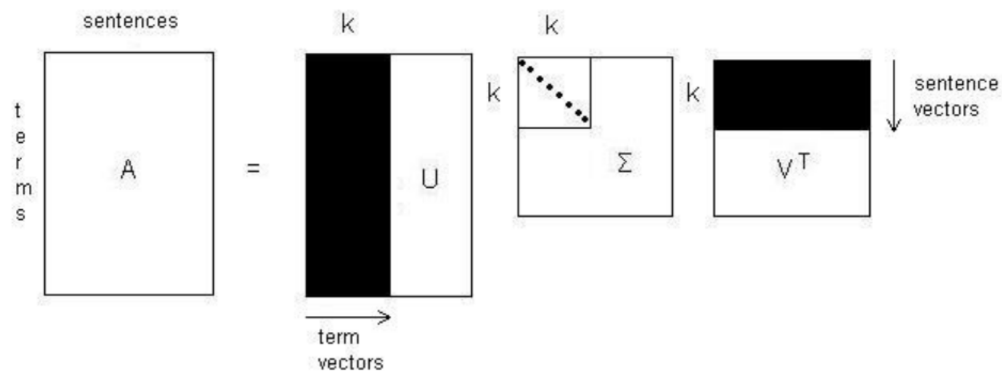

Fig. 1: SVD Decomposition

Salient and recurring word patterns are likely to be captured and represented by a singular vector. The corresponding eigen value indicates the degree of importance of the pattern. Sentences containing this pattern will be projected along this vector and the sentence that best represents this pattern will have the largest component along this vector.

When using LSA for summarization, it is assumed that a document deals with a small number of representative concepts. Under this assumption, the sentence best representing each concept is considered to be relevant to the summary of the document. In Steinberger et al [20], two major disadvantages of this method are identified: (1) it is difficult to fix the number of concepts that need to be considered and (2) a sentence that explains many concepts reasonably well but is not the *best* sentence for any concept does not get selected.In order to overcome this, they propose a modification of LSA where, each sentence is assigned a score based on the extent to which it explains a concept, weighted by the importance (eigen value) of that concept. Mathmatically, if $\mathbf{v}$ is the sentence vector in the concept space and σ_i is the eigen value of the i^{th} concept, the score of a sentence is given by

$$score(\mathbf{v}) = \sqrt{\sum_i \sigma_i^2 v_i^2} \tag{2}$$

3.2 Vector-Space Models

The LSA method makes use of a term-sentence matrix. Different vector space models can be used to represent sentences in a term space. Some of these are briefly dicussed below:

Binary features This is the basic bag-of-words model where, each term in the document is a dimension in the space. A sentence has a feature value of 1 for a word if it contains a word and 0 otherwise.

Term Frequency Sometimes it is desirable to capture the number of occurences of a word in a particlar sentence. This can be used as a feature value in the term-sentence matrix.

TF.IDF The previous models assume that all words are of equal importance when characterizing sentences. However, a word that occurs in several sentences may not be particularly useful in distinguishing one sentence from the rest. To capture this, a variant of TF.IDF, commonly used in Information Retrieval, can be used. In place of IDF, we use the inverse sentence frequency, calculated as follows:

$$IDF(w) = \log_2 \frac{n_w}{N} \tag{3}$$

where, n_w is the number of sentences in which w occurs and N is the total number of sentences in the document.

Unigram Probabilities We may not wish for very common words to be given high importance just because it occurs only a few times in the document, for example, the use of words like furthermore, additionally or albeit. In this approach, the term frequency of a words has been scaled inversely with its unigram probability as follows:

$$A(w, s) = TF(w, s) \times \log(-\log(p(w))) \tag{4}$$

where $TF(w, s)$ is the term frequency of the word w in the sentence s and $p(w)$ is the unigram probability of w.

3.3 Diversity Awareness

Observe that the nature of score in equation (2) allows for the selection of multiple sentences explaining the same concept if it has a high eigen value. While more than one sentence may be required, it is usually desirable to enforce some diversity among the sentences selected to ensure that even less important topics get covered. This principle is, in fact, used in the evaluation of Maximal Marginal Relevance [3] as a metric for diversity. Drawing inspiration from MMR, we attempt to improve the diversity of the selected sentences by using a score that is a weighted combination of the relevance of a sentences, as captured by the LSA score and its diversity with respect to the sentences already selected as follows:

$$score(\mathbf{v}^{(i)}) = (1 - \lambda)\sqrt{\sum_k \sigma_k^2 \left(v_k^{(i)}\right)^2} + \lambda \max_{v^{(j)} \in S} < v^{(i)}, v^{(j)} > \tag{5}$$

3.4 Evaluation

Automatic text summarizers are usually evaluated by comparison against model summaries usually provided by humans. However, it is difficult to obtain a consensus on what constitutes a model summary. There is also little inter-annotator agreement when humans are asked to score summaries.

A number of metrics have been used in literature to evaluate automatic summaries. The most popular standard among these is ROUGE [22]. ROUGE (Recall-Oriented Understudy of Gisting Evaluation) is a set of metrics used to compare system generated and model summaries. We use the ROUGE-n metric, n-gram recall based metric, calculated as follows -

$$\text{ROUGE-}n(s) = \frac{\sum_{r \in R} < \phi(r), \phi(s) >}{\sum_{r \in R} < \phi(r), \phi(r) >} \tag{6}$$

where R is a set of reference (model) summaries and $\phi(d)$ is the vector of n-grams of document d.

One of the disadvantages of ROUGE is that it does not look for a sentence-level mapping between the generated and model summaries. To address this, we also considered the following evaluation scheme.

For every sentence in the generated summary, we consider its closest match (in terms of Jaccard similarity of unigrams or other n-grams) in the model summary. The mean of the similarities of each sentence in the generated summary to its closest match is taken as the score of the summary. The Jaccard similarity of a sentence to its closest match will be 1 if it is present in the model summary. We note that this evaluation measure is stricter than the usual ROUGE-n class of measures in the sense that we expect every sentence in the generated summary to be relevant.

4 Experiments

4.1 Datasets

Most of our experiments have been conducted on the TIPSTER dataset [16], which consists of scientific papers that appeared in Association for Computational Linguistics (ACL) sponsored conferences. The dataset consists of 183 documents; the abstract of a paper is taken to be a model summary of the paper. Note that the model summary in this case would not be extractive.

Small scale evaluation is also done on the Reuters dataset (Reuters-21578, Distribution 1.0), a popular collection of news documents that appeared on the Reuters news wire in 1987 and a comments-oriented blog summarization dataset [10]. In both these cases, we take the ground-truth to be the extractive summary generate the Microsoft Word Summarizer. Large scale evaluation could not be performed on these as summaries had to be obtained manually using Microsoft Word 2007.

4.2 Construction of Term-Sentence Matrix

To create the term-sentence matrix, we first the parse the data which is available in XML format. During this process, all the URLs were replaced with the word *url*. Next, we lemmatize all the words in the sentences of the parsed output using the pattern.en package in Python and remove all stop-words. We have used a list of approximately 500 stop-words downloaded off the Web for this purpose.

4.3 Estimation of Unigram Probabilities

The unigram probabilities of content-words are obtained from the Brown and Gutenberg corpora. We stem the words in the corpora before using them in order to obtain more reliable counts. Also, to overcome the problem of zero-counts, we use Good-Turing smoothing.

4.4 Results on TIPSTER Dataset

Table 1 depicts the performance of a representative sample from the TIPSTER dataset for different vector space models on both ROUGE-1 and the proposed evaluation metric.

λ	Binary		Unigram Pbty		TF		TF.IDF	
	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1
0	0.02	0.01	0.12	0.41	0.01	0.03	0.02	0.03
0.1	0.02	0.01	0.12	0.41	0.04	0.26	0.02	0.03
0.2	0.05	0.26	0.12	0.41	0.09	0.33	0.04	0.1
0.3	0.27	0.41	0.12	0.43	0.11	0.36	0.1	0.31
0.4	0.28	0.49	0.12	0.37	0.12	0.39	0.1	0.31
0.5	0.28	0.49	0.11	0.33	0.11	0.37	0.12	0.4
0.6	0.11	0.41	0.13	0.37	0.09	0.37	0.1	0.33
0.7	0.11	0.41	0.13	0.37	0.09	0.37	0.11	0.41
0.8	0.11	0.41	0.12	0.3	0.09	0.37	0.1	0.36
0.9	0.11	0.41	0.12	0.3	0.09	0.36	0.1	0.39
1	0.11	0.41	0.12	0.3	0.09	0.36	0.1	0.39

Table 1: Comparison of performance of vector-space models on ROUGE-1 and our score on a TIPSTER sample

When $\lambda = 1$, the pure LSA score is being used to select sentences. We observe that for most cases, the score increases, reaches a maximum and then decreases. For high values of λ, due to insufficient diversity, not all concepts are well-covered, leading to lower n-gram recall and hence lower values of scores. For very low λs, since the LSA does not play a major role, most of the selected sentences are not very relevant. Since the size of the summary is fixed, this also lowers the scores. Also, note that the performance on our score is much worse than that on ROUGE-1. This is expected as we impose stricter constraints and the reference summaries used a re not extractive.

To compare the different vector-space models and different values of λ, mean scores over the entire TIPSTER dataset were computed. These are presented in Table 2.

λ	Binary	Unigram Pbty	TF	TF.IDF
0	0.07522	0.09467	0.082802	0.091868
0.1	0.12626	0.14879	0.14467	0.14918
0.2	0.22407	0.23731	0.245	0.22275
0.3	0.3222	0.32824	0.34214	0.30313
0.4	0.40857	0.38764	0.39346	0.35077
0.5	0.44676	0.41341	0.4172	0.37577
0.6	0.45764	0.41962	0.42758	0.3906
0.7	0.46159	0.41874	0.42879	0.39956
0.8	0.46692	0.42352	0.43242	0.41192
0.9	0.46357	0.42879	0.43154	0.41324
1	0.46209	0.43011	0.43077	0.41379

Table 2: Mean ROUGE-1 scores on TIPSTER dataset

For better visualization, the variation of ROUGE-1 scores for TIPSTER dataset have been plotted in Figure 2.

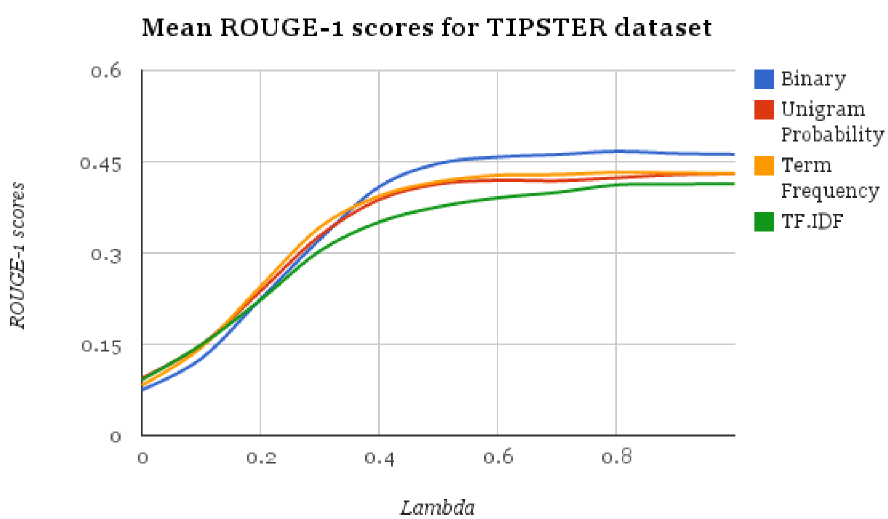

Fig. 2: Mean ROUGE-1 scores on TIPSTER dataset

The use of TF-IDF results in a relatively bad performance. This can be explained by the fact that the words that are strongly indicative of the topic are likely to occur in many sentences and hence get low IDF scores. In contrast, sometimes the non-content words such as furthermore or additionally which are common in English occur rather infrequently in the text and are hence assigned higher IDF scores.

Surprisingly, aspects such as the rarity of a word (which would be captured in unigram probabilities) seem to have little effect. We had initially assumed that by weighting the term frequency using a decreasing function of unigram probability, non-dictionary words (which are probably proper nouns and hence the subject of discussion) would receive a higher weightage. This can be seen by comparing the values in the third and fourth columns of Table 2.

Also, for binary and term-frequency spaces, the best mean score is obtained at $\lambda = 0.8$. This indicates that a combination of LSA and a diversity aware term is more suitable than the use of LSA alone.

Similar trends have been observed with our scores as well.

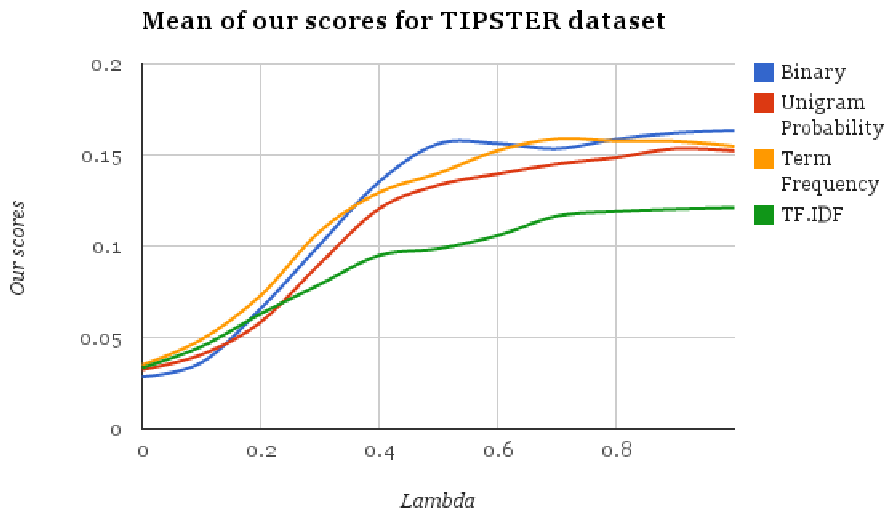

Fig. 3: Mean of our scores on TIPSTER dataset

4.5 Results on Blog and Reuters Datasets

Performance on representative files from blog and Reuters datasets are given in tables 3 and 4 respectively.

λ	Binary		Unigram Pbty		TF		TF.IDF	
	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1
0	0.48	0.57	0.34	0.55	0.4	0.57	0.52	0.61
0.1	0.48	0.57	0.36	0.6	0.44	0.58	0.54	0.63
0.2	0.46	0.59	0.38	0.65	0.47	0.68	0.51	0.68
0.3	0.49	0.64	0.41	0.66	0.48	0.73	0.57	0.77
0.4	0.56	0.79	0.46	0.73	0.55	0.79	0.59	0.83
0.5	0.59	0.81	0.52	0.8	0.6	0.82	0.57	0.83
0.6	0.6	0.83	0.55	0.84	0.58	0.86	0.57	0.86
0.7	0.57	0.86	0.55	0.88	0.55	0.84	0.53	0.87
0.8	0.62	0.88	0.58	0.89	0.57	0.86	0.53	0.86
0.9	0.6	0.87	0.55	0.87	0.57	0.86	0.51	0.83
1	0.58	0.86	0.6	0.89	0.58	0.87	0.49	0.83

Table 3: Comparison of performance of vector-space models on ROUGE-1 and our score on a blog sample

λ	Binary		Unigram Pbty		TF		TF.IDF	
	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1	Our Score	ROUGE-1
0	0.49	0.56	0.5	0.54	0.4	0.51	0.62	0.63
0.1	0.49	0.74	0.5	0.54	0.5	0.54	0.62	0.63
0.2	0.49	0.74	0.51	0.52	0.5	0.54	0.51	0.6
0.3	0.39	0.72	0.51	0.59	0.4	0.57	0.51	0.7
0.4	0.39	0.72	0.42	0.49	0.4	0.57	0.61	0.79
0.5	0.41	0.71	0.42	0.49	0.5	0.7	0.61	0.79
0.6	0.41	0.7	0.42	0.62	0.61	0.83	0.61	0.87
0.7	0.42	0.7	0.4	0.66	0.61	0.83	0.5	0.79
0.8	0.4	0.68	0.5	0.7	0.5	0.7	0.4	0.72
0.9	0.4	0.68	0.5	0.7	0.5	0.7	0.4	0.72
1	0.4	0.68	0.5	0.7	0.4	0.68	0.31	0.61

Table 4: Comparison of performance of vector-space models on ROUGE-1 and our score on a Reuters sample

It is observed that similar trends are seen on the Blog and Reuters datasets as well. However, the actual scores here are significantly higher. This is because, the models summaries used for these datasets are those obtained from Microsoft Word Summarizer, which are truly extractive in nature as opposed to the abstractive summaries in the TIPSTER dataset. Moreover, an abstract is not necessarily a good summary of a research paper, so it is not surprising that the performance was worse on the TIPSTER dataset.

The visualization of the ROUGE-1 scores of the Blog and Reuters datasets aggregated over a representative sample of the documents are given in figures 4 and 5.

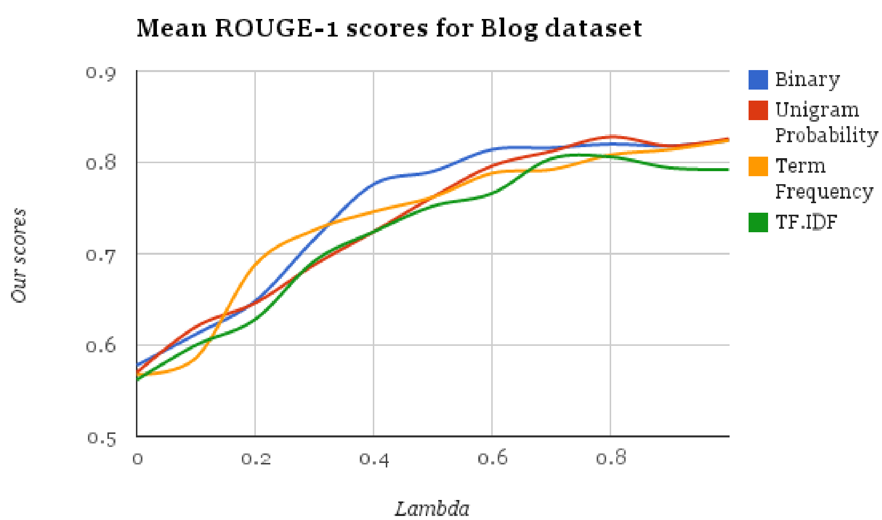

Fig. 4: Mean ROUGE-1 scores on a subset of the Blog dataset

5 Conclusions and Future Work

In this project, we have demonstrated the importance of incorporating diversity in any LSA-based summarization techniques. We have also compared the different vector-spaces that can be used for performing LSA. We have also empirically shown that binary features works better than more complex representations like

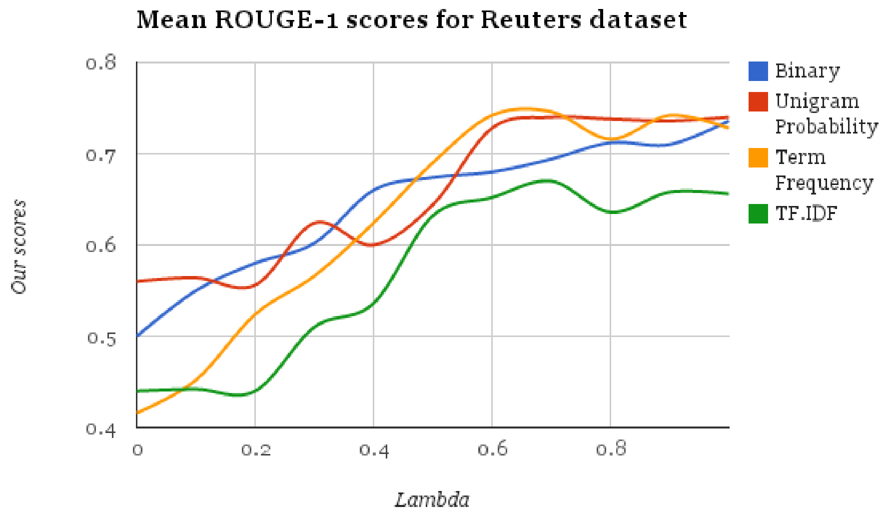

Fig. 5: Mean ROUGE-1 scores on a subset of the Reuters dataset

TF or TF.IDF or unigram probability weighting. We have also intuitively explained why these trends could be expected. We have also proposed a more rigorous evaluation metric which uses sentence level comparison rather than simply treating the summary as a bag of n-grams. Future work in this area could examine interesting ways of augmenting the term-sentence matrix with surface-level features such as the position of the sentence within the document. Care must be taken to avoid incorporation of other surface-level features like sentence length, similarity with the title which are linearly dependent the existing term features. Another direction could be to incorporate a score such as the one presented as a feature in a supervised setting.

References

1. Regina Barzilay and Michael Elhadad. Using lexical chains for text summarization. *Advances in automatic text summarization*, pages 111–121, 1999.
2. A. Bookstein, S. T. Klein, and T. Raita. Detecting content-bearing words by serial clustering—extended abstract. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 319–327, New York, NY, USA, 1995. ACM.
3. Jaime Carbonell and Jade Goldstein. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '98, pages 335–336, New York, NY, USA, 1998. ACM.
4. J. Cowie, K. Mahesh, S. Nirenburg, and R Zajaz. Minds-multilingual interactive document summarization. In *Working Notes of the AAAI Spring Symposium on Intelligent Text Summarization*, AAAI '98, pages 131–132, 1998.
5. Hal Daumé, III and Daniel Marcu. Bayesian query-focused summarization. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, ACL-44, pages 305–312, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.
6. Scott Deerwester, Susan T. Dumais, George W. Furnas, Thomas K. Landauer, and Richard Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.
7. David Kirk Evans, David Kirk Evans, and David Kirk Evans. Abstract identifying similarity in text: Multi-lingual analysis for summarization, 2005.
8. R.A. Garcia-Hernandez and Y. Ledeneva. Word sequence models for single text summarization. In *Advances in Computer-Human Interactions, 2009. ACHI '09. Second International Conferences on*, pages 44–48, Feb 2009.
9. Vishal Gupta and Gurpreet Singh Lehal. A survey of text summarization extractive techniuqes. *Journal of emerging technologies in web intelligence*, 2(3), August 2010.

10. Meishan Hu, Aixin Sun, and Ee-Peng Lim. Comments-oriented document summarization: understanding documents with readers' feedback. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 291–298, New York, NY, USA, 2008. ACM.
11. Khosrow Kaikhah. Automatic text summarization with neural networks. In *Proceedings of second international Conference on intelligent systems*, IEEE '04, pages 40–44, 2004.
12. Chris Kedzie, Kathleen McKeown, and Fernando Diaz. Summarizing disasters over time.
13. Julian Kupiec, Jan Pedersen, and Francine Chen. A trainable document summarizer. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 68–73, New York, NY, USA, 1995. ACM.
14. Chin-Yew Lin and Eduard Hovy. From single to multi-document summarization: A prototype system and its evaluation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 457–464, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.
15. H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, April 1958.
16. Inderjeet Mani, David House, Gary Klein, Lynette Hirschman, Therese Firmin, and Beth Sundheim. The tipster summac text summarization evaluation. In *EACL*, pages 77–85, 1999.
17. Elke Mittendorf and Peter Schäuble. Document and passage retrieval based on hidden markov models. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '94, pages 318–327, New York, NY, USA, 1994. Springer-Verlag New York, Inc.
18. Joel Larocca Neto, Alex Alves Freitas, and Celso A. A. Kaestner. Automatic text summarization using a machine learning approach. In *Proceedings of the 16th Brazilian Symposium on Artificial Intelligence: Advances in Artificial Intelligence*, SBIA '02, pages 205–215, London, UK, UK, 2002. Springer-Verlag.
19. H Gregory Silber and Kathleen F McCoy. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496, 2002.
20. Josef Steinberger and Karel Jeek. Using latent semantic analysis in text summarization and summary evaluation. In *In Proc. ISIM 04*, pages 93–100, 2004.
21. Meng Wang, Xiaorong Wang, and Chao Xu. An approach to concept oriented text summarization. In *Proceedings of ISCIT05*, ISCIT05, pages 1290–1293, 2005.
22. Chin yew Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.