Unsupervised Text Summarization Using Sentence Embeddings

Aishwarya Padmakumar
aish@cs.utexas.edu
ap44694

Akanksha Saran
asaran@cs.utexas.edu
as65859

Abstract

Dense vector representations of words, and more recently, sentences, have been shown to improve performance in a number of NLP tasks. We propose a method to perform unsupervised extractive and abstractive text summarization using sentence embeddings. We compare multiple variants of our systems on two datasets, show substantially improved performance over a simple baseline, and performance approaching a competitive baseline.

1. Introduction

Dense vector representations of words [21, 24] have seen many successful applications in NLP [3, 30, 28]. More recently, dense vector representations of sentences have been shown to be successful at tasks such as predicting textual similarity and textual entailment, and in sentiment classification [11, 33]. In this project, we propose a method to use sentence embeddings, specifically those trained by Weieting et al. [33] to detect paraphrases for text summarization.

Text Summarization is the process of condensing source text into a shorter version, preserving its information content and overall meaning. With the explosion of data available on the Web in the form of unstructured text, efficient methods of summarizing text are important, due to the inability of people to assimilate vast quantities of information. Text summarization techniques typically employ various mechanisms to either identify highly relevant sentences in the text or remove redundant phrases/sentences [5]. We propose to cluster sentences projected to a high dimensional vector space to identify groups of sentences that are semantically similar to each other and select representatives from these clusters to form a summary.

Summarization tasks can be categorized in a number of ways. One of these is the length of the summary, which can broadly be classified into methods that aim to create a headline or a set of keywords, and methods that aim to generate a short but coherent sequence of sentences. We tackle the second type of task. Our method is unsupervised, which is important given that most datasets for this type of summarization task are relatively small in size.

Another broad classification of summarization techniques is into extractive and abstractive summarization. Extractive summarization methods identify relevant sentences from the original text and string them together to form a summary. Abstractive summarization methods are those that can generate summary sentences that are not present in the original text[5]. We propose both an extractive and abstractive summarization paradigm, both of which are applicable to any sentence embedding. We test our approach using two state of the art sentence embedding techniques - skip thought vectors [11] and paraphrastic sentence embeddings [33].

2. Related Work

Text summarized is a fairly well-studied problem in literature right from the late 1950s. One of the first attempts to solve this problem came from Luhn et al [17] which used used high-frequency words present in the document to score a sentence for relevance. Over the years, several techniques have been applied for solving this problem including some recent attempts using neural networks [26, 16, 22, 14]. [26, 16, 22] use various forms of attention based encoder decoder models to generate keyword/headline style summaries. In contrast, our method is used for generating multi-sentence summaries, where sentences in the summary are expected to deal with distinct semantic concepts. [14] use an auto-encoder to learn a low dimensional embedding of a paragraph and could be potentially be used for summarization because as the length of the document increases, the system is likely to generate a condensed version of the original document. However, this has not actually been tested in this manner on text summarization.

This idea of clustering sentences in a high-dimensional space has also been used for text summarization in the past [19, 20, 2]. However, those systems used TF-IDF representations of sentences (which are only applicable in a multi-document summarization system) instead of sentence embeddings. Another class of vector space based methods use Latent Semantic Indexing [6] to identify sentences that best explain latent concepts in the document [29, 32].

Another class of text summarization methods is graph based approaches. These range from modelling the document as a similarity graph with sentences as nodes [12] to lexical chains [1, 27]. We compare our methods against one such method that operates on a word-level graphical representation of a document [9].

There have also been a some earlier attempts at performing supervised extractive summarization, which use a variety of features such as content and title words, sentence location, sentence length, upper case words and cue phrases, to classify sentences in a text to be summary or non-summary sentences [23, 13, 4].

3. Background

3.1. Skip Thought Vectors

This work [11] aims to encode sentences in a vector space using an RNN with LSTM [10]. Sentence embeddings are learned in a manner similar to the skip-gram method for learning word embeddings. The basic idea behind this is that vectors of sentences should be predictive of the context surrounding those sentences, which in this case is represented by the vectors of the previous and next sentences.

3.2. Paraphrastic Sentence Embeddings

Most technqiues that combine word embeddings to form sentence embeddings are general purpose, learned in an unsupervised manner, and not targeted towards any specific task. This work [33] aims to learn how to combine word embeddings to obtain sentence embeddings that satisfy the property that sentences that are paraphrases of each other are embedded near each other in the vector space. This is done in a supervised manner using known paraphrases. The authors compare different technqiues for combining word embeddings and test the learned embeddings on prediction of textual similarity and entailment, and in sentiment classification. They find that averaging word embeddings learned in a supervised manner performs best for prediction of textual similarity and entailment. We use this embeddings from this model in experiments using *paragram embeddings*.

4. Approach

We propose an unsupervised text summarization approach by clustering sentence embeddings trained to embed paraphrases near each other. Clusters of sentences are then converted to a summary by selecting a representative from each cluster. To select a representative from each cluster, we propose an extractive and an abstractive method. The extractive method simply chooses that sentence from the text whose embedding is the nearest, in terms of Euclidean distance, to the centroid of the cluster. In the abstractive method, a decoder is trained to decode embeddings into sentences. We used a recurrent neural network with long short term memory [10] to decode embeddings into sentences. Specifically, we used the decoder from [31]. Although that work used the model to obtain natural language descriptions of videos, the decoder itself simply converts vectors into sentences. The source of the vector is irrelevant. An important point to note is that given an encoder, we can potentially generate an infinite amount of training data for the decoder by encoding any available raw text using the encoder.

When representing sentences in a high-dimensional vector space, the goal is typically to directly or indirectly embed sentences such that sentences close in meaning are embedded near each other in the vector space. Thus, sentences that form a cluster in the vector space are likely to be close in meaning to each other. We exploit this assumption to perform summarization. Since sentences that form a cluster in the vector space are likely to be close in meaning to each other, it is sufficient to retain one representative from each such cluster to form a summary.

5. Experiments

5.1. Datasets

The standard datasets for testing summarization techniques are the DUC datasets. Due to complications with obtaining these datasets, we used the following two datasets to test our methods in varied domains.

The Tipster dataset [18] is a collection of scientific papers that appeared in Association for Computational Linguistics (ACL) sponsored conferences. The dataset consists of 183 documents and the abstract of a paper is taken to be a model summary of the paper. The metadata in this dataset was poor and even after removing unwanted sections such as the references in an automated preprocessing step, some of these had to be removed manually.

The Opinosis dataset [9] is a collection of sentences extracted from user reviews grouped by topic. In total there are 51 such topics with each topic having approximately 100 sentences (on average). The topics are drawn from Tripadvisor.com (hotels), Amazon.com (products) and Edmunds.com (cars). The dataset comes with about 4-5 gold standard summaries per topic.

5.2. Baseline

Most text summarization methods report results on the DUC datasets, making it difficult for direct comparison with our method. We compare our method with the MEAD[25] and Opinosis [9] methods as baselines.

MEAD[25] is an extractive technique for multiple document summarization based on cluster centroids. It uses a collection of the most important words from the whole

cluster to select the best sentences for summarization. By default, the scoring of sentences in MEAD is based on 3 parameters - minimum sentence length, centroid and position in text.

Opinosis [9] is a graph based method for unsupervised text summarization evaluated on the Opinosis dataset. This framework is well-suited to capture highly redundant opinions/text to generate concise abstractive summaries. A textual graph is first constructed that represents the text to be summarized, words in the text forming nodes of the graph and adjacent words generating directed edges between nodes. Unique properties of this graph (redundancy capture, collapsable structures, gapped subsequence capture) are used to explore and score various subpaths that help in generating candidate abstractive summaries. The graph structure naturally captures redundancies and collapsible structures which outperforms the MEAD system [25]. The authors of this work show that more than 60% of the sentences generated as part of their summaries are judged as human-generated, making 40% of generated sentences non-readable (as picked by human evaluators). graph emphasizes too much on the surface order of words. As a result, it cannot group sentences at a deep semantic level.

5.3. Variants of the system

We implemented the extractive and abstractive techniques using two types of sentence embeddings - skip thought vectors [11] and paragram embeddings [33].

For training the decoder, we attempted a few variants. We vary the vocabulary of the decoder and its training set. We tried both restricted (domain specific) and generic vocabularies and training sets. In a restrcited training set, the training data for the decoder only consisted of encoded sentences from the summarization corpus itself. The restricted vocabulary only contained words used in these sentences.

For the more general training set, we combined sentences from the two summarization corpora used (discussed in section 5.1) and the Brown corpus [7, 8]. The general vocabulary contained words from all these sentences. We did not include words that occurred less that 3 times in either the restricted or generic vocabulary.

We also attempted two methods for clustering - K-means and Mean-shift clustering. We experimented with a range of parameter settings for each of these. More details on this are included in the appendix.

5.4. Evaluation

As is standard for text summarization, we evaluate our systems using ROUGE [34]. ROUGE is based on n-gram co-occurrence between machine summaries and human summaries. In our experiments, we report results with ROUGE-1 and ROUGE-2 metrics. ROUGE-1 and ROUGE-2 have been shown to have most correlation with human summaries [15] and higher order ROUGE-N scores (N ¿1) estimate the fluency of summaries.

6. Results and discussion

The scores of the different systems on ROUGE-1 on the Tipster dataset can be seen in Table 2. We do not have scores from the baseline methods on this dataset. We observe that the extractive systems in general perform substantially better than the abstractive ones, contrary to our expectations. On observing the output, we noticed that the decoder tended to generate a fair number of ⟨UNK⟩ tokens. This is possibly because this is a dataset of scientific papers, which has a number of words that do not occur frequently enough in the dataset itself, or in the Brown corpus, leading to poor parameter estimates in the decoder.

We also observe that the systems using Paragram embeddings have a higher precision than those using skip-thought embeddings. However, systems using skip-thought embeddings perform better on recall and also on F-score. One possible reason for the low precision in this dataset is the presence of a number of rare words, which could be encoded to ⟨UNK⟩ by both embeddings. We also do not see a clear trend between the different types of abstractive systems that is consistent across clustering methods. On comparing clustering methods, K-means seems to perform better than Mean-shift clustering.

Table 3 lists the ROUGE-2 scores of the systems on the Tipster dataset. We notice that the absolute scores are much lower than the corresponding scores. This is natural since the ROUGE-2 metric is based on 2-gram overlap with gold summaries, as opposed to 1-gram overlap in ROUGE-1. The trends across different types of systems are otherwise quite similar to ROUGE-1.

The scores of the different systems and the two baselines on ROUGE-1 on the Opinosis dataset can be seen in Table 4. This is important because the recall can naively be increased by including more sentences in the summary. We observe that our systems significantly outperform the simpler MEAD baseline, but do not outperform the more competitive Opinosis baseline, although we approach close to it. Some sample summaries from various systems on this dataset can be seen in Table 1.

In this dataset, our abstractive systems outperform the extractive systems as expected. This is a simpler dataset in terms of vocabulary, as they are reviews written by general consumers, which possibly helps us learn a better decoder. Also, as expected, the best performing abstractive system is the one using the restricted vocabulary but the generic training data. We expected a restricted vocabulary to improve performance as it reduces the number of parameters in the decoder, and generic training data would help as it is larger in size.

Also, we observe a similar trend to the Tipster dataset

when comparing types of embeddings. Again the Paragram embeddings perform better on precision, and skip-thought embeddings perform better on recall. However, as the vocabulary is less difficult, the improved precision of Paragram embeddings is sufficient to result in a higher F score. Another reason why we believe high precision is helpful in this dataset is that the gold standard summaries are very short, often just one or two sentences.

We observe similar trends in ROUGE-2, as seen in Table 5. Again, we come close to the Opinosis, but do not manage to outperform it, although we do overtake the simpler MEAD baseline by a comfortable margin.

Between the two datasets, the ROUGE-1 scores are comparable but we do better on ROUGE-2 on Opinosis. Since higher N-gram based ROUGE metrics tend to measure fluency, the simpler vocabulary of the Opinosis dataset probably results in the increased ROUGE-2 score.

One factor we believe could be holding back our abstractive systems is that we could not spend much time tuning the hyperparameters of the deep network. Since the original network in [31] is tuned for input vectors of a different size, it is possible that we could perform better using different hyperparameters. Another possible method to improve our systems would be to incorporate an additional check or score to decide whether a cluster of sentences is sufficiently important to be represented in the summary.

7. Conclusion

We demonstrate how the clustering of sentence embeddings can be used to perform both extractive and abstractive text summarization. We compare several variants of our proposed system on two datasets. We show improved performance over a simple baseline and performance approaching a competitive baseline system. We believe that our system could outperform the baseline system with additional hyperparameter tuning or an additional relevance check on summary sentences. A follow-up study of our work would be on how to sequence the cluster centroids which form the summary, to result in maximum fluency.

References

[1] R. Barzilay and M. 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] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, and P. Kuksa. Natural language processing (almost) from scratch. *J. Mach. Learn. Res.*, 12:2493–2537, Nov. 2011.

[4] J. M. Conroy and D. P. O'leary. Text summarization via hidden markov models. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 406–407. ACM, 2001.

[5] D. Das and A. F. Martins. A survey on automatic text summarization. Literature Survey for the Language and Statistics II course at CMU, 2007.

[6] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman. Indexing by latent semantic analysis. *JOURNAL OF THE AMERICAN SOCIETY FOR INFORMATION SCIENCE*, 41(6):391–407, 1990.

[7] W. N. Francis. A standard sample of present-day english for use with digital computers. Report to the U.S Office of Education on Cooperative Research Project No. E-007. Brown University, Providence RI., 1964.

[8] W. N. Francis and H. Kucera. Frequency analysis of english usage: Lexicon and grammar. Houghton Mifflin, 1982.

[9] K. Ganesan, C. Zhai, and J. Han. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 340–348. Association for Computational Linguistics, 2010.

[10] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[11] R. Kiros, Y. Zhu, R. R. Salakhutdinov, R. Zemel, R. Urtasun, A. Torralba, and S. Fidler. Skip-thought vectors. In *Advances in Neural Information Processing Systems*, pages 3276–3284, 2015.

[12] C. Kruengkrai and C. Jaruskulchai. Generic text summarization using local and global properties of sentences. In *Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence*, WI '03, pages 201–, Washington, DC, USA, 2003. IEEE Computer Society.

[13] J. Kupiec, J. Pedersen, and F. 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] J. Li, M.-T. Luong, and D. Jurafsky. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057*, 2015.

[15] C.-Y. Lin and E. Hovy. Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 71–78. Association for Computational Linguistics, 2003.

[16] K. Lopyrev. Generating news headlines with recurrent neural networks. arXiv:1512.01712, 2015.

[17] H. P. Luhn. The automatic creation of literature abstracts. *IBM J. Res. Dev.*, 2(2):159–165, Apr. 1958.

[18] I. Mani, D. House, G. Klein, L. Hirschman, T. Firmin, and B. Sundheim. The tipster summac text summarization evaluation. In *EACL*, pages 77–85, 1999.

Ground Truth	Video camera is great. Very easy to use. Video quality is excellent.
Paragram-kmeans-extractive	as for the video camera it's a nice addition considering it's an mp3 player first and foremost, i think people are going to go into this and think maybe i can replace my old video, no . even the built, in video camera is very good .
Paragram-kmeans-abstractive-generic-generic	The video camera is great and ⟨UNK⟩ but a big camera is ⟨UNK⟩ even if the picture in.
Skipthought-kmeans-extractive	it takes video and has a really cool radio feature on it, according to the kids .
Skipthought-kmeans-abstractive-generic-generic	⟨UNK⟩ the video camera is just good and has fun fun for every day ..

Table 1. Some sample summaries generated by various systems and ground truth for the same document

Extractive / Abstractive	Embedding	Clustering method	Vocabulary	Training data	Precision	Recall	F score
Extractive	Paragram	K-means	-	-	**0.4141**	0.2321	**0.2693**
Extractive	Paragram	Mean Shift	-	-	0.2931	0.2615	0.2459
Abstractive	Paragram	K-means	Restricted	Restricted	0.2187	0.2855	0.2255
Abstractive	Paragram	K-means	Restricted	Generic	0.2309	**0.2925**	0.2346
Abstractive	Paragram	K-means	Generic	Generic	0.2251	0.2906	0.2299
Abstractive	Paragram	Mean Shift	Restricted	Restricted	0.1434	0.274	0.1638
Abstractive	Paragram	Mean Shift	Restricted	Generic	0.1486	0.2823	0.1675
Abstractive	Paragram	Mean Shift	Generic	Generic	0.1527	0.2889	0.172
Extractive	Skip-thought	K-means			**0.3366**	0.2862	**0.2822**
Extractive	Skip-thought	Mean Shift			0.1403	**0.3574**	0.1796
Abstractive	Skip-thought	K-means	Restricted	Restricted	0.1974	0.2572	0.2082
Abstractive	Skip-thought	K-means	Restricted	Generic	0.2115	0.2681	0.2193
Abstractive	Skip-thought	K-means	Generic	Generic	0.2211	0.2808	0.229
Abstractive	Skip-thought	Mean Shift	Restricted	Restricted	0.0835	0.2974	0.1205
Abstractive	Skip-thought	Mean Shift	Restricted	Generic	0.0901	0.2984	0.1255
Abstractive	Skip-thought	Mean Shift	Generic	Generic	0.097	0.3188	0.1326

Table 2. ROUGE-1 results on Tipster dataset

[19] K. McKeown and D. R. Radev. Generating summaries of multiple news articles. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '95, pages 74–82, New York, NY, USA, 1995. ACM.

[20] K. R. McKeown, J. L. Klavans, V. Hatzivassiloglou, R. Barzilay, and E. Eskin. Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, pages 453–460, Menlo Park, CA, USA, 1999. American Association for Artificial Intelligence.

[21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases

Extractive / Abstractive	Embedding	Clustering method	Vocabulary	Training data	Precision	Recall	F score
Extractive	Paragram	K-means			**0.0843**	**0.0454**	**0.0526**
Extractive	Paragram	Mean Shift			0.041	0.0418	0.0369
Abstractive	Paragram	K-means	Restricted	Restricted	0.0231	0.0311	0.0242
Abstractive	Paragram	K-means	Restricted	Generic	0.0234	0.0309	0.0241
Abstractive	Paragram	K-means	Generic	Generic	0.0216	0.03	0.0224
Abstractive	Paragram	Mean Shift	Restricted	Restricted	0.0105	0.0268	0.0134
Abstractive	Paragram	Mean Shift	Restricted	Generic	0.0107	0.0255	0.0133
Abstractive	Paragram	Mean Shift	Generic	Generic	0.012	0.0271	0.0143
Extractive	Skip-thought	K-means			**0.0758**	0.0611	**0.0603**
Extractive	Skip-thought	Mean Shift			0.0231	**0.0671**	0.0318
Abstractive	Skip-thought	K-means	Restricted	Restricted	0.0161	0.0236	0.0178
Abstractive	Skip-thought	K-means	Restricted	Generic	0.0162	0.0226	0.0176
Abstractive	Skip-thought	K-means	Generic	Generic	0.0189	0.0271	0.0202
Abstractive	Skip-thought	Mean Shift	Restricted	Restricted	0.0069	0.0276	0.0103
Abstractive	Skip-thought	Mean Shift	Restricted	Generic	0.0072	0.0222	0.0087
Abstractive	Skip-thought	Mean Shift	Generic	Generic	0.0077	0.0298	0.0112

Table 3. ROUGE-2 results on Tipster dataset

and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119, 2013.

[22] R. Nallapati, B. Zhou, C. N. dos santos, aglar Gulehre, and B. Xiang. Abstractive text summarization using sequence-to-sequence rnns and beyond. arXiv:1602.06023, 2016.

[23] J. L. Neto, A. A. Freitas, and C. 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.

[24] J. Pennington, R. Socher, and C. D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[25] D. R. Radev, H. Jing, and M. Budzikowska. Centroid-based summarization of multiple documents: sentence extraction, utility-based evaluation, and user studies. In *Proceedings of the 2000 NAACL-ANLP Workshop on Automatic summarization*, pages 21–30. Association for Computational Linguistics, 2000.

[26] A. M. Rush, S. Chopra, and J. Weston. A neural attention model for sentence summarization. In *EMNLP*, 2015.

[27] H. G. Silber and K. F. McCoy. Efficiently computed lexical chains as an intermediate representation for automatic text summarization. *Computational Linguistics*, 28(4):487–496, 2002.

[28] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, 2013.

[29] J. Steinberger and K. Jeek. Using latent semantic analysis in text summarization and summary evaluation. In *In Proc. ISIM 04*, pages 93–100, 2004.

[30] J. Turian, L. Ratinov, and Y. Bengio. Word representations: A simple and general method for semi-supervised learning. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, ACL '10, pages 384–394, Stroudsburg, PA, USA, 2010. Association for Computational Linguistics.

[31] S. Venugopalan, H. Xu, J. Donahue, M. Rohrbach, R. J. Mooney, and K. Saenko. Translating videos to natural language using deep recurrent neural networks. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 1494–1504, 2015.

[32] M. Wang, X. Wang, and C. Xu. An approach to concept oriented text summarization. In *Proceedings of ISCIT05*,

Extractive / Abstractive	Embedding	Clustering method	Vocabulary	Training data	Precision	Recall	F score
Baseline - MEAD					0.0916	**0.4932**	0.1515
Baseline - Opinosis					**0.4482**	0.2831	**0.3271**
Extractive	Paragram	K-means			**0.4323**	0.1347	0.2003
Extractive	Paragram	Mean Shift			0.3443	0.1598	0.2127
Abstractive	Paragram	K-means	Restricted	Restricted	0.3543	0.2107	0.2598
Abstractive	Paragram	K-means	Restricted	Generic	0.3588	0.2004	0.2531
Abstractive	Paragram	K-means	Generic	Generic	0.3537	0.1963	0.249
Abstractive	Paragram	Mean Shift	Restricted	Restricted	0.2691	**0.2734**	0.2667
Abstractive	Paragram	Mean Shift	Restricted	Generic	0.3001	0.2644	**0.2767**
Abstractive	Paragram	Mean Shift	Generic	Generic	0.2819	0.2531	0.2626
Extractive	Skip-thought	K-means			0.2601	**0.2613**	0.2516
Extractive	Skip-thought	Mean Shift			**0.2695**	0.2556	**0.2517**
Abstractive	Skip-thought	K-means	Restricted	Restricted	0.1998	0.2075	0.2005
Abstractive	Skip-thought	K-means	Restricted	Generic	0.2473	0.2478	0.2435
Abstractive	Skip-thought	K-means	Generic	Generic	0.2569	0.2485	0.2479
Abstractive	Skip-thought	Mean Shift	Restricted	Restricted	0.2068	0.2	0.1988
Abstractive	Skip-thought	Mean Shift	Restricted	Generic	0.2564	0.2393	0.2421
Abstractive	Skip-thought	Mean Shift	Generic	Generic	0.2649	0.2402	0.2458

Table 4. ROUGE-1 results on the Opinosis dataset

ISCIT05, pages 1290–1293, 2005.

[33] J. Wieting, M. Bansal, K. Gimpel, and K. Livescu. Towards universal paraphrastic sentence embeddings. *arXiv preprint arXiv:1511.08198*, 2015.

[34] C. yew Lin. Rouge: a package for automatic evaluation of summaries. pages 25–26, 2004.

Appendix

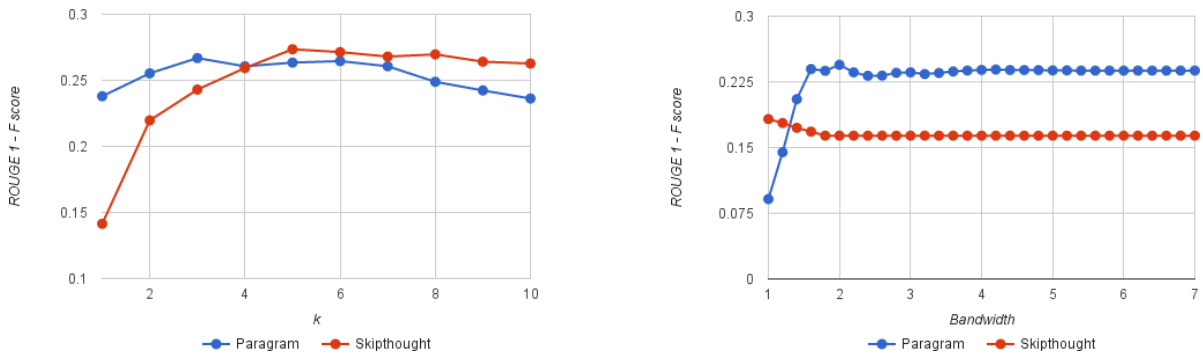
Here we report results on validation data (20% of the available data) for both Tipster and Opinosis datasets. The experiments reported here were used in choosing hyper parameter values for k-means (number of clusters) and mean-shift clustering (bandwidth) for the test data (Tables 2-5).

Fig.1 depicts how ROUGE-1 scores (F-scores) vary as (a) number of clusters increase for k-means clustering and (b) bandwidth increases for mean-shift clustering on the Tipster data. Fig.2 depicts a similar trend for the Opinosis dataset. We show these variations for both Paragram and Skip-thought embeddings. We choose hyper-parameter values corresponding to the highest Rouge-1 scores obtained on these graphs. We report results in the report above on the test data for these hyper-parameters. Fig.1(a) and 2(a) show that the Rouge-1 scores for k-means have varying curves whereas for mean-shift in Fig.1(b) and 2(b) the curves are mostly flat after a point. The values shown for mean-shift clustering are on a fine-grained scale. Prior to running this experiment, we experiment with a coarser range of bandwidth values but found good performance only on this narrow range. The paragram and skip-thought embeddings show a similar trend with respect to each other though the associated scores vary based on the dataset.

Fig.3 depicts change in ROUGE-2 scores (F-scores) as (a) number of clusters increase for k-means clustering and (b) bandwidth increases for mean-shift clustering on the Tipster data. Fig.4 depicts a similar trend for the Opinosis dataset. We do not choose hyper-parameters from these experiments since we do not see as significant a change in ROUGE-2 scores as compared to ROUGE-1. Here again, mean-shift curves in Fig.3(b) and Fig.4(b) are mostly flat.

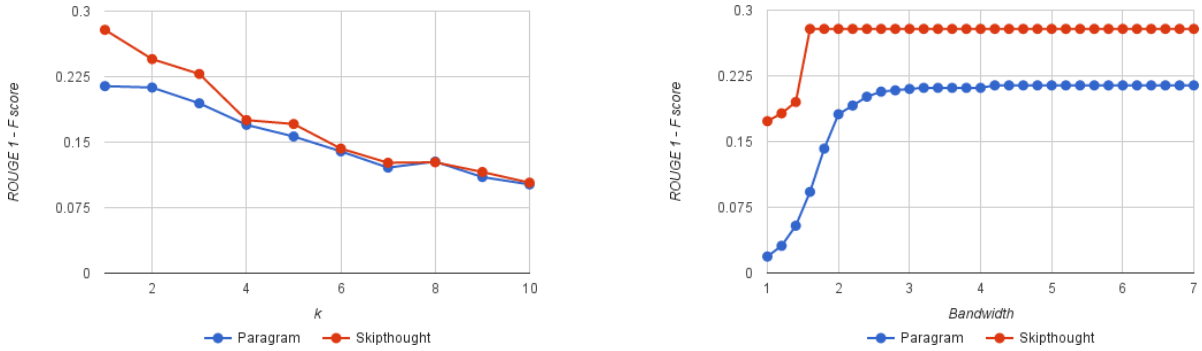
Extractive / Abstractive	Embedding	Clustering method	Vocabulary	Training data	Precision	Recall	F score
Baseline - MEAD					0.0184	**0.1058**	0.0308
Baseline - Opinosis					**0.1416**	0.0853	**0.0998**
Extractive	Paragram	K-means			0.1106	0.0292	0.0449
Extractive	Paragram	Mean Shift			0.0691	0.0302	0.0405
Abstractive	Paragram	K-means	Restricted	Restricted	0.1177	0.0601	0.0776
Abstractive	Paragram	K-means	Restricted	Generic	**0.1186**	0.0549	0.0731
Abstractive	Paragram	K-means	Generic	Generic	0.1152	0.0528	0.0708
Abstractive	Paragram	Mean Shift	Restricted	Restricted	0.0836	**0.0745**	0.0762
Abstractive	Paragram	Mean Shift	Restricted	Generic	0.1013	0.0744	**0.0836**
Abstractive	Paragram	Mean Shift	Generic	Generic	0.0851	0.0646	0.0715
Extractive	Skip-thought	K-means			0.0604	0.0527	0.0533
Extractive	Skip-thought	Mean Shift			0.0606	0.0515	0.0528
Abstractive	Skip-thought	K-means	Restricted	Restricted	0.0353	0.0326	0.0326
Abstractive	Skip-thought	K-means	Restricted	Generic	0.0678	**0.0562**	0.0596
Abstractive	Skip-thought	K-means	Generic	Generic	0.0702	0.0561	**0.0602**
Abstractive	Skip-thought	Mean Shift	Restricted	Restricted	0.0355	0.031	0.0317
Abstractive	Skip-thought	Mean Shift	Restricted	Generic	0.0686	0.0536	0.0585
Abstractive	Skip-thought	Mean Shift	Generic	Generic	**0.0716**	0.0548	0.06

Table 5. ROUGE-2 results on Opinosis dataset



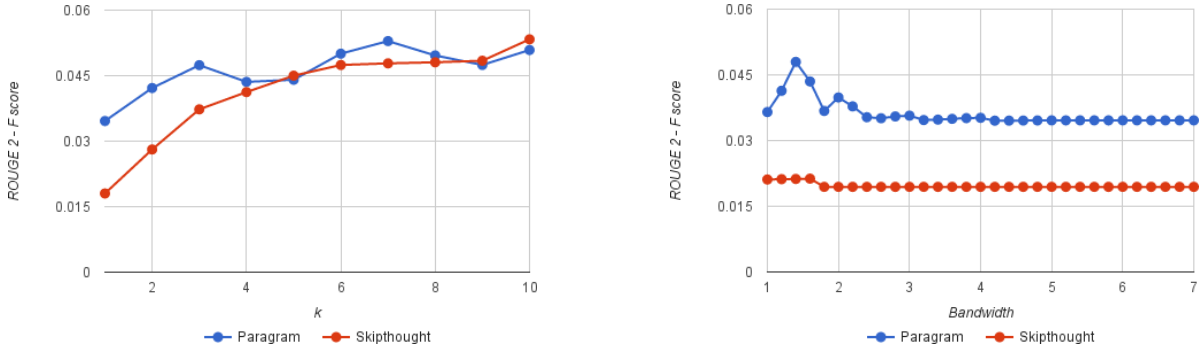
(a) K-means (b) Mean-shift

Figure 1. Performance of the Tipster dataset evaluated using ROUGE-1 scores.



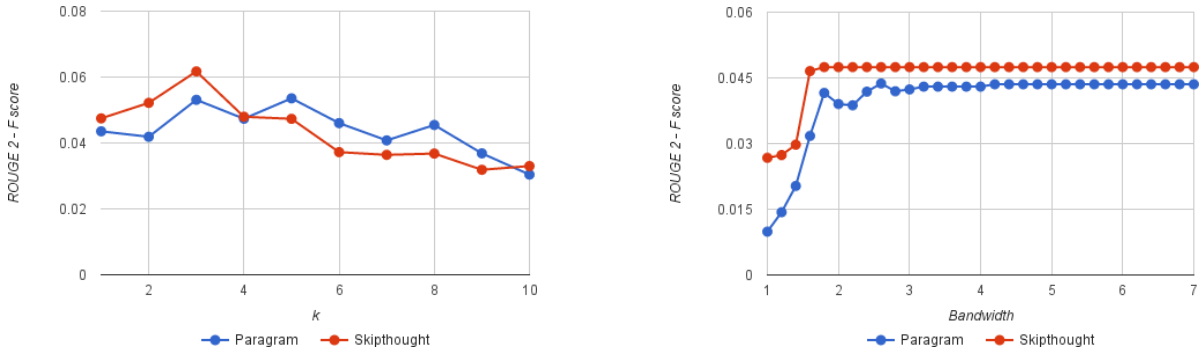
(a) K-means (b) Mean-shift

Figure 2. Performance of the Opinosis dataset evaluated using ROUGE-1 scores.



(a) K-means (b) Mean-shift

Figure 3. Performance of the Tipster dataset evaluated using ROUGE-2 scores.



(a) K-means (b) Mean-shift

Figure 4. Performance of the Opinosis dataset evaluated using ROUGE-2 scores.