Automatic Summarization of Danish Text

Xuemin Duan
University of Copenhagen
npm838@alumni.ku.dk

Abstract

Most studies in automatic text summarization have only focused on English but are limited to other languages, especially Danish. To our knowledge, there is no published work related to the neural network model for Danish automatic text summarization. In this paper, we build the first neural extractive summarization system on the Danish language dataset and present the performance of different models over the test set to be a possible reference for future work. Specifically, we implemented five models and analyzed their results to find out whether there is a statistically significant difference between the results of the models and which groups of results are different from each other. The experimental results show that BertSum achieved the best performance.

1 Introduction

The rapid growth and popularization of the Internet in recent years leads to the explosive growth of text information. People would access a huge amount of text on the Internet every day, such as news, blogs, tweets, and academic articles, which have also heightened the need to quickly extract the main ideas from the original document. Automatic Text Summarization (ATS) plays an important role in addressing the need.

ATS approaches can be divided into extractive, abstractive, and hybrid (El-Kassas et al., 2021). The extractive approach identifies the important sentences in the original document and composes them into a summary. The abstractive approach uses advanced natural language processing algorithms to generate summary word by word that may differ from the original sentences.

In recent years, researchers have shown an increased interest in the ATS system and developed several automatic summarization datasets. For example, (Hermann et al., 2015) and (See et al., 2017) established the CNN / Daily Mail dataset for summarization, and (Liu, 2019) proposed a simple variant of BERT for extractive summarization.

However, most studies in the field of ATS have only focused on English but are limited to other languages (Nguyen and Daumé III, 2019), especially Danish. To our knowledge, there is little published work on Automatic Danish Text Summarization, only one work related to the establishment of the first Danish summarization dataset (DaNewsroom) was published in 2020 (Varab and Schluter, 2020). Varab implemented some baseline models on DaNewsroom, but the performances of neural networks models are still lacking.

This paper aims to contribute to this research gap by exploring several different baselines and neural networks methods on the Danish summarization dataset collaborating with Infomedia, a leading media intelligence powerhouse in the Nordics. To our knowledge, there are only non-neural-based extractive summarization systems for the Danish language before this paper. With our work, the contributions are as follows:

- We are the first ones to build the neural extractive summarization system on the Danish language dataset.
- INFOMEDIA[1] established a Danish automatic summarization dataset. We implemented several different models on this dataset and present their performances as the reference for future work.
- We analyzed whether there is a statistically significant difference between the results of the models, and find out exactly which groups of results are different from each other.

2 Related Work

Automatic text summarization systems can be classified as extractive and abstractive based on the text

[1] https://infomedia.org/

summarization approach, single-document, and multi-document based on the input size, monolingual, multilingual, and cross-Lingual based on the summary language.

Much of the literature since the mid-2000s focused on traditional machine learning algorithms. In (Erkan and Radev, 2004), Erkan and Radev proposed a Graph-based Lexical Centrality method to compute the relative importance of sentences. (Chen and Song, 2009) discussed a Vector Space Model based on the clustering algorithm. (Shetty and Kallimani, 2017) utilize K-means clustering based on cosine similarity to obtain a summary. In the same vein, (Mallick et al., 2019) present a modified graph-based TextRank approach to calculate the similarity between pairs of sentences instead of the regular TF-IDF. Similarly, (Mohd et al., 2020) implemented an extractive summarization system, using a distributional semantic model to capture the semantics of the text as a feature, using the K-means algorithm to cluster the semantically similar sentences, then using a rank algorithm to select an important sentence.

More recent attention has focused on neural automatic summarization. The development of neural network architectures and the collection of large-scale data has enabled the transition from heuristic-based systems to end-to-end deep neural models(Kryściński et al., 2019). In (Zhang et al., 2018), zhang treat extractive summarization task as a sequence labeling task, proposed a latent variable extractive model with a Bidirectional Long Short-Term Memory Network (Bi-LSTM) (Schuster and Paliwal, 1997) as encoder and a LSTM (Hochreiter and Schmidhuber, 1997) as decoder to predict sentence labels. (Zhou et al., 2018) present an end-to-end neural network framework that combined two main steps of sentence scoring and sentence selection for extracting sentences. (Liu and Lapata, 2019) proposed a general framework based on BERT to apply text summarization. In (Liu, 2019), Liu proposed a BertSum model for extractive summarization and achieve a state-of-the-art performance. By drawing on the BertSum model of Liu, (Zhong et al., 2020) present a MatchSum framework based on BERT to optimize the results of BertSum.

2.1 Danish Text Summarization

To our knowledge, there is very little work related to the Danish text summarization has been published in recent years. (Varab and Schluter, 2020) present a first large-scale Danish language summarization dataset and some unsupervised baseline systems.

3 Dataset

In this paper, we focused on the Danish summarization dataset established by INFOMEDIA which includes 27,849 articles with manually gold summaries. After removing some exception examples, the dataset is left with 24,569 articles. The articles of the dataset are collected from higher education institutions in Denmark and the gold summaries are manually summarized in the Danish language. Figure 3 shows the distribution of the total number of sentences of articles and summaries. The average total number of sentences in gold summaries is 3.79 and that of articles is 38.43.

4 Methodology

This section outlines the models and methods that were utilized in our paper. We use the LEAD and ORACLE as baselines and implement LexRank, K-means, BertSum, and MatchSum models on the INFOMEDIA dataset.

4.1 LEAD

LEAD is one of the most common baselines in automatic text summarization. It believes that the sentence position is an important feature and the first k sentences often contain the main topics of the article (Grenander et al., 2019). This method directly selects the first k sentences of the article to compose the candidate summary.

4.2 ORACLE

ORACLE is often referred to as an upper-bound performance for automatic text summarization systems. In this paper, we adopt two algorithms to implement ORACLE by selecting sentences that would maximize the ROUGE-1 score with respect to the gold summary. Ideally, we should calculate the ROUGE scores of all sentence combinations and then select the best one, but it would cause a combinatorial explosion since there are too many combinations. For solving this problem, the previous method selects candidate summaries through the greedy algorithm. We always select the sentence with the highest ROUGE-1 score as the next candidate sentence to combine with the selected candidate sentences to compose the summary.

The latter method extracts the candidate summary based on LexRank. We choose the top 10 sentences ranking by LexRank as candidate sentences. Then calculate the ROUGE scores of all combinations of candidate sentences and select the combination of the highest score as the candidate summary. It can be thought of as the upper bound of the performance of the LexRank.

4.3 LexRank

TextRank(Mihalcea and Tarau, 2004) is a graph-based method for extracting the most important k sentences, borrows the idea from PageRank(Page et al., 1999). Google's PageRank is an algorithm for ranking web pages through their links. Similarly, TextRank ranks sentences through the cosine similarity between different sentences and computing the ranking score using the iterative calculation of PageRank.

However, this method also has a number of drawbacks. The article may have many similar sentences, all of which have high ranking scores but contain similar information. Because of this, the output will lose sub-important information and include redundant data. For getting rid of the above problem, (Erkan and Radev, 2004) introduced LexRank.

LexRank set a threshold to filter sentences that are not very similar to the current sentence. For example, if the threshold is 0.3, only the sentences with a cosine similarity of more than 0.3 would be considered relevant. Then use the power iteration method to calculate eigenvector centrality to find out the most important sentence. It consists of the following four steps:

1. Split article into sentences and add them as vertices in the graph.
2. Compute intra-sentence cosine similarity and use them as edges of the graph.
3. Iterate the graph-based ranking algorithm until convergence.(Mihalcea and Tarau, 2004)
4. Sort sentences based on their ranking score.

The ranking score:

$$p(u)=\frac{d}{N}+\sum_{v\in adj[u]}\frac{(1-d)cosine(u,v)p(v)}{\sum_{z\in adj[v]}cosine(z,v)} \quad (1)$$

Where N is the total number of the nodes in the graph. $cosine(u,v)$ is the cosine similarity score of the sentence pair (u,v). $p(u)$ is the ranking score of article u, $adj[u]$ is the set of articles that are adjacent to u.

In (Erkan and Radev, 2004), they use the bag-of-words model to represent each sentence. Differs from Erkan, we choose to use the contextualized word embeddings from BERT to calculate cosine similarity of the pair of sentences.

4.4 K-means

K-means clustering algorithm is a commonly used clustering algorithm based on Euclidean distance. In our paper, we set the number of centroids, k, equals to the number of sentences of the candidate summary. First randomly select k centroid as the initial centroid. Then calculate the distance between each sentence and each centroid, and assign each sentence to the nearest centroid. Iteratively recalculate the centroid of each cluster until the centroid does not change. Finally, the sentences closest to each centroid are selected and composed together as candidate summaries.

4.5 Fine-tuning BERT for extractive summarization

(Liu, 2019) proposed a novel method, Bertsum, to extract summaries using modified BERT. The architecture of the BertSum model is shown in Figure 1. BertSum treats the extractive automatic text summarization task as a binary classification task. It builds several different summarization layers on the Bert to get the predicted score $\hat{Y}_i$ after getting the output from BERT. Differs from the original input of the BERT model which only adds one [CLS] token at the start of the whole input document, for assigning 0-1 label to each sentence of input, BertSum inserts an external [CLS] token in front of each sentence. Then using the Binary Classification Entropy of the predicted label $\hat{Y}_i$ and the gold label Y_i as the loss of this model.

The gold summaries are written manually in an "abstractive" way rather than an "extractive" way, so we cannot obtain gold labels from gold summaries directly. For getting the gold label, (Liu, 2019) chose to calculate the F_1 score of ROUGE-1 and ROUGE-2 of each sentence, then assign 1 to the top-k sentences and 0 to others as the gold label. In our paper, instead of selecting the top-k sentences, we use a different greedy algorithm to obtain a gold label. First, we select a sentence with the highest ROUGE-1 F_1 score. Then we calculate the score of the combination of each remaining sentence and the selected sentences. At each step, we select the next sentence which can generate the highest score combined with all selected sentences.

4.6 Two-stage MatchSum summarization framework

(Zhong et al., 2020) proposed a two-stage extract-then-match framework to optimize summarization results by using a siamese-BERT architecture based on the intuition that the best candidate summary is not always composed of best k sentences. They formulate the extractive automatic text summarization task as a semantic text matching problem. They fine-tuned BERT based on the principle idea that the better candidate summary should be semantically closer to the original article, and the gold summary should be semantically closest to the original article. In this paper, we leverage this novel method to optimize our results from LexRank. At the first stage, we prune sentences for addressing the combinatorial explosion problem. We selected the top-6 ranking score sentences from the original article through LexRank, then generated all combinations of the pre-selected sentences as the input of MatchSum. At the second stage, we fine-tuned BERT by calculating the cosine similarity between the gold summary and the original article, and the cosine similarity between the pre-selected candidate summaries and the original articles, which is shown in Figure 2.

5 Experiments

5.1 Dataset

We evaluated our models on the INFOMEDIA dataset, which contains 24,569 article-summary pairs, and split this dataset into 15,725 training, 3,930 validation, and 4,914 test examples.

5.2 Evaluation Metrics

ROUGE (Recall-Oriented Understudy for Gisting Evaluation) is a common evaluation method for automatic text summarization task proposed by (Lin, 2004). This measure counts the number of overlapping units such as n-gram between the gold summary and the candidate summary.

We use the F_1 score of ROUGE-1, ROUGE-2, and ROUGE-L for evaluation.

5.3 Implementation Details

We leverage the Danish SpaCy tokenizer to do sentence split and word segmentation for all of our models.

Baselines We use LEAD and ORACLE as the baseline models. For LEAD, we select the first three and four sentences of the articles respectively as the candidate summaries. For ORACLE, we predict the candidate summary by selecting the best four sentences one by one based on the gold summary.

LexRank and K-means We use danlp library to obtain BERT embedding and employ a grid search on the training set for optimizing the parameter combination of summary sentences number and threshold. For LexRank, we found the sentences number of 4 and threshold of 0.1 can give the best results. For K-means, we got the number of the sentence of 4 achieved the best performance.

BertSum

In this paper, we implemented two different BertSum models. BertSum uses the 'danish-bert-botxo' version of BERT as the encoder and two different summarization layers respectively as the decoder. The first architecture adds a linear layer on the BERT outputs and uses a sigmoid function to get the predicted score. The second architecture adds an inter-sentence Transformer layer on top of the Bert and still uses the sigmoid function to get output. All models are trained for 15,000 steps on a single Tesla-P100-16G GPU. Model checkpoints are saved and evaluated on the validation set every 1,500 steps. We use grid search to optimize hyperparameters of accumlate count and warmup steps. For BertSum with linear layer, we got the number of sentences of 3, accumulate count of 2, and warm-up steps of 2,400 can generate the best results. For BertSum with transformer layer, we got the number of sentences of 4, accumulate count of 16, and warmup steps of 1,000 achieved the best performance. Accumulate count refers to the number of updates steps to accumulate before performing a backward pass.

MatchSum We use the 'danish-bert-botxo' version of BERT to implement MatchSum model, the Adam optimizer with warming-up as the optimization algorithm. The initial learning rate is set to $2e^{-3} \cdot min(step^{-0.5}, step \cdot wm^{-1.5})$, follows as (Vaswani et al., 2017). The 5-fold cross validation was performed over 19,655 article-summary pairs (15,725 for training, 3,930 for validation). For tuning the hyper-parameters, we employed a grid search, optimizing for ROUGE-1 F_1 score, on the training set. Finally, we found that the batchsize

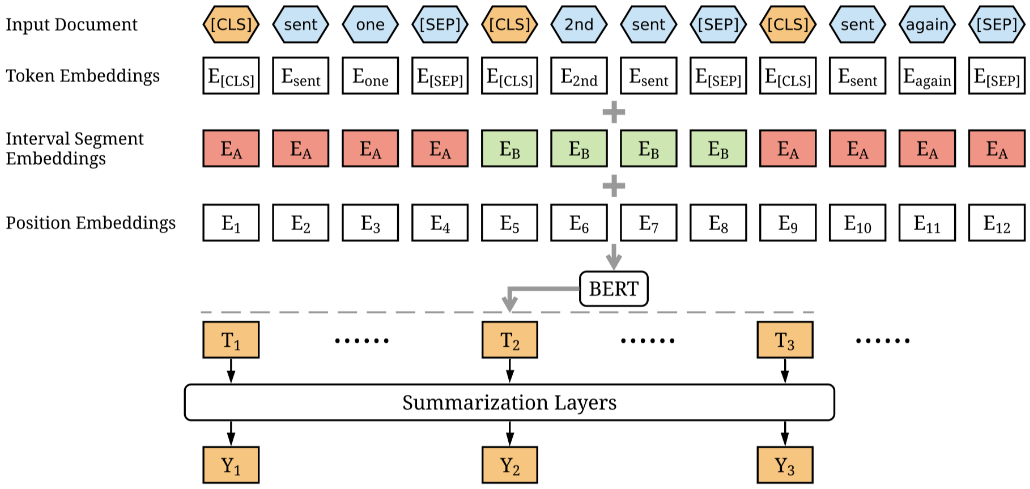

Figure 1: The overview architecture of the BERTSUM model.(Liu, 2019)

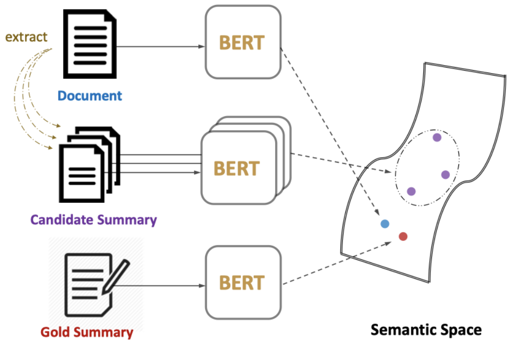

Figure 2: MATCHSUM framework (Zhong et al., 2020).

of 4 and the accumulate count of 8 can give us the best result. The model is trained for 4 epochs on a Tesla-P100-16G GPU for about 40 hours.

5.4 Experimental results

The experimental results are shown in Table 1. The first section contains several baselines. Since it is not possible to calculate the ORACLE score of all combinations of sentences to reach the global optimum, we selected the candidate sentences one by one through the greedy algorithm to reach a local optimum. We calculated the ORACLE-MatchSum+LexRank score based on the top-6 sentences through LexRank, so it is relatively high.

LEAD is often seen as the lower abound of the automatic text summarization system. As illustrated in Table 1, LexRank with the number of sentences of 3, MatchSum over LexRank, and all BertSum models beat the LEAD baselines. And all BertSum model beats LexRank and K-means by a large margin. BertSum with linear layer achieved the best performance. MatchSum as an optimization method significantly improved the results of LexRank. We didn't implement the MatchSum+BertSum model, but based on its performance on LexRank, we have reasonable evidence to believe that MatchSum+BertSum would lead to better results.

6 Discussion

In this section, we discussed three questions as follows:

1. What are the advantages and disadvantages of using the INFOMEDIA dataset to train our automatic text summarization models?

2. What are the shortcomings of LexRank in terms of the candidate summaries it generated?

3. Are there significant differences between the results of our different models?

6.1 Pros and Cons of INFOMEDIA Dataset

The articles in the INFOMEDIA dataset are collected by Infomedia.org and the gold summaries of the articles are manually summarized, which can be relatively representative of the level of human intelligence. The automatic summarization system is designed to summarize the summary as closely as possible as a human expert, so manually gold summaries can help us train the models and evaluate the results more meaningfully.

Besides that, the manually gold summaries also come with a downside. Figure 3 shows that the

Model	**ROUGE-1**	**ROUGE-2**	**ROUGE-L**
LEAD - 3	28.10	9.21	25.13
LEAD - 4	29.41	9.80	26.45
ORACLE - greedy algorithm	33.64	14.15	30.64
ORACLE - MatchSum + LexRank - 4	36.79	15.34	32.15
LexRank + tf-idf word embedding (num=3)	25.21	8.71	21.41
LexRank + Bert word embedding (num=3)	28.27	9.29	24.95
LexRank + Bert word embedding (num=4)	31.18	12.17	28.32
K-means + Bert word embedding (num=4)	26.35	8.91	22.92
BertSum + Transformer Layer (num=3)	32.44	11.48	28.12
BertSum + Transformer Layer (num=4)	35.05	13.57	30.45
BertSum + Linear Layer (num=3)	**35.90**	**14.29**	**31.15**
BertSum + Linear Layer (num=4)	35.18	13.58	30.21
MatchSum + LexRank - 4	32.09	13.23	29.27

Table 1: F_1-score ROUGE on the INFOMEDIA test set. NUM=3 stands for candidate summary composed of three sentences, NUM=4 stands for candidate summary composed of four sentences.

number of sentences in the gold summaries ranges from 0 to 10 instead of a fixed number. The automatic summarization models need to set a sentence number for extracting candidate summaries in advance. But we cannot use the number of sentences in the gold summary of the article, because we wouldn't know this number in the real application. These gold summaries of the different number of sentences pose a question that how to determine the number of sentences in the candidate summary.

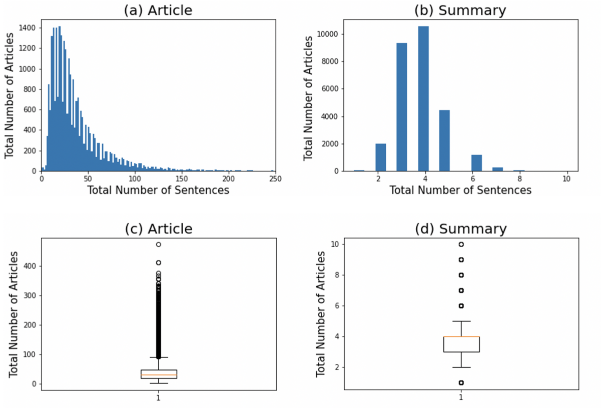

Figure 3: Distribution of the total number of sentences of articles and summaries.

For addressing this problem, we try to find a relation between sentence number and article length to predict the appropriate number of sentences in candidate summaries for our model. However, our analysis results turn out that there is no direct relation between them. Long articles do not necessarily have long gold summaries. Therefore, in this paper, we artificially chose 3 and 4 as the number of sentences in candidate summaries, which may affect the performance of our models because of possible inconsistencies with the number of sentences in the gold summaries.

In addition to the different number of sentences of summaries the long text article also is an obstacle to training our model. For example, the maximum input sequence length of BERT is restricted to 512, but most of our articles' lengths range from 600 to 800. For fine-tuning Bert, we need to do additional modifications to the model.

Except for the above, the limited size of the dataset also poses a challenge to us. The INFOMEDIA dataset includes 24,569 article-summary pairs, which is relatively small for a neural network model. For dealing with it, we use 5 fold cross validation to train our MatchSum model.

So far this paper has focused on the extractive automatic text summarization models, let us now consider whether the abstractive automatic text summarization model would be a better fit to work on this dataset. I argued that the abstractive method is more suitable for this dataset based on the performance of ORACLE. ORACLE is often seen as the upper bound of the ATS system. From Table 1, we can see that the maximum ROUGE-1 score of ORACLE is 36.79, which suggests that even the performance of the best extractive model is unlikely to exceed this result. Therefore, the abstractive automatic text summarization model may have greater potential to achieve better performance. In future work, we need to pay more attention to the abstractive ATS system.

6.2 Shortcomings of LexRank

In analyzing the candidate summaries generated by lexrank, we found that LexRank tends to fall into the trap of generating summaries that contain important information but have a high redundancy. LexRank scores all the sentences of the article and then selects the top-k sentences to compose a summary.

One example of the first 15 sentences ranked by LexRank of an article is included in Appendix. We can see that the important information of the gold summary includes ’new innovation network Danish Healthtech’, ’Danish educational institutions’, and ’Aalborg University’. The candidate summary composed of the first three sentences covers the most important information, Danish Healthtech, but ignores the other several secondary important information. Although these three sentences all revolve around ’Danish Healthtech’, they provide no new information, resulting in high redundancy.

Lexrank’s scoring is reasonable, and these first top-k sentences did contain the most important concept. However, it is ignored that the purpose of the ATS system is to generate a summary that covers all main topics and does not include redundant or repeated data. We can find that if the candidate summary is composed of top-4 sentences, then it covers the second main topic, ’Danish educational institutions’, since the fourth sentence contains it. And the remaining main topic is contained in the fifteenth sentence. Therefore, if LexRank can reduce the redundancy of candidate sentences by taking only one similar sentence and giving others a lower weight, it may gain a better performance.

6.3 Difference between Results

For exploring whether there is a statistically significant difference between the means of the results of the models, we use ANOVA (Analysis of Variance) (Miller Jr, 1997) to analyze it. We proposed the following hypotheses:

H_0 There is no difference between the means of results of the models.

H_1 The means of results of the models are not all equal.

We run an ANOVA on the results of the LexRank, K-Means, MatchSum+LexRank, BertSum+Transformer Layer, and BertSum+Linear Layer models. We select the models in Table 1 that achieves the best performance compared with the same model with different sentences number to do analysis.

Model1	Model2	p-value	reject
K-Means	BertSumLL	0.001	True
K-Means	LexRank	0.001	True
K-Means	MatchSum	0.001	True
K-Means	BertSumTF	0.001	True
BertSumLL	LexRank	0.001	True
BertSumLL	MatchSum	0.001	True
BertSumLL	TBertSumTF	0.0012	True
LexRank	MatchSum	0.001	True
LexRank	BertSumTF	0.001	True
MatchSum	BertSumTF	0.001	True

Table 2: The results of the multiple comparison of means using Tukey HSD. Family-Wise Error Rate (FWER) equals to 0.05. K-Means stands for ‘K-means + Bert word embedding (num=4)’. BertSumLL stands for ’BertSum + Linear Layer (num=3)’. LexRank stands for ’LexRank + Bert word embedding (num=4)’. BertSumTF stands for ’BertSum + Transformer Layer (num=3)’.

Then we got the f-value of 524.5522 and the p-value of 0. Since the p-value is less than 0.05, we reject the null hypothesis at a 95% level of confidence, which means that we have sufficient evidence to say that at least one of the means of the results is different from the others.

In order to determine exactly which groups of results are different from each other, we applied Tukey’s multiple comparison test. The p-values of each pair of models are shown in Table 2. From this table, we can see that all null hypothesis have been rejected, which means that there is a statistically significant difference between the means of all pairs of models.

For seeing the difference of results more clearly, we drew a Kernel Density Estimation (KDE) plot (Terrell and Scott, 1992) as shown in Figure 3. From this figure, we can see that these curves are all different from each other. Meanwhile, the curve of ’BertSum+Linear Layer’ is relatively similar to that of ’BertSum+Transformer Layer’ and the curve of ’LexRank’ is relatively similar to that of ’MatchSum+LexRank’. Therefore, although the results of all pairs of models are statistically significantly different, the differences between the results of relatively similar models are smaller than others.

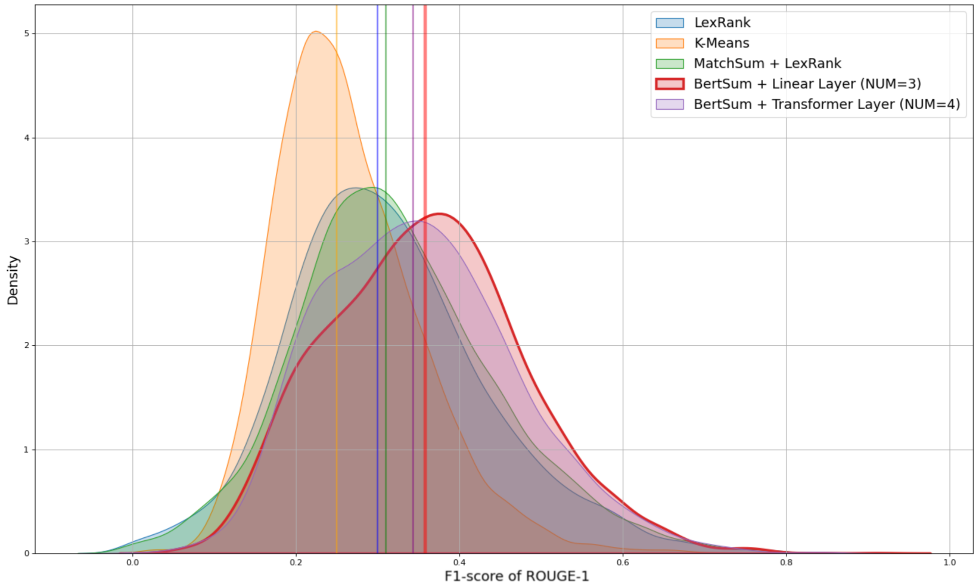

Figure 4: Kernel Density Estimation plot. The x-axis is the F_1-score of ROUGE-1. The y-axis is an estimate of a probability density function of the F_1-score of ROUGE-1 using kernel density estimation.

7 Conclusion

We presented the first neural extractive automatic text summarization system on the Danish language dataset. We show the performance of different models over the test set to be the possible reference for future work. We conduct a statistical test to find out whether there is a statistically significant difference between the results of the models and find out exactly which groups of results are different from each other. Experimental results show that BertSum achieved the best performance and MatchSum significantly improved the result of LexRank. We believe that MatchSum with BertSum has a great potential to bring a better result.

Acknowledgement

I'd like to express my deepest thanks to Maria Maistro, my supervisor, for her consistent encouragement and expert guidance. I'd also like to express my gratitude to Joselene Marques for her assistance in data acquisition and other supports. The work was collaborating with INFOMEDIA, thanks INFOMEDIA for establishing the dataset and providing a laptop for my work.

References

Mingzhen Chen and Yu Song. 2009. Summarization of text clustering based vector space model. In *2009 IEEE 10th International Conference on Computer-Aided Industrial Design & Conceptual Design*, pages 2362–2365. IEEE.

Wafaa S El-Kassas, Cherif R Salama, Ahmed A Rafea, and Hoda K Mohamed. 2021. Automatic text summarization: A comprehensive survey. *Expert Systems with Applications*, 165:113679.

Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of artificial intelligence research*, 22:457–479.

Matt Grenander, Yue Dong, Jackie Chi Kit Cheung, and Annie Louis. 2019. Countering the effects of lead bias in news summarization via multi-stage training and auxiliary losses. *arXiv preprint arXiv:1909.04028*.

Karl Moritz Hermann, Tomas Kocisky, Edward Grefenstette, Lasse Espeholt, Will Kay, Mustafa Suleyman, and Phil Blunsom. 2015. Teaching machines to read and comprehend. *Advances in neural information processing systems*, 28:1693–1701.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Wojciech Kryściński, Nitish Shirish Keskar, Bryan McCann, Caiming Xiong, and Richard Socher. 2019.

Neural text summarization: A critical evaluation. *arXiv preprint arXiv:1908.08960*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*, pages 74–81.

Yang Liu. 2019. Fine-tune bert for extractive summarization. *arXiv preprint arXiv:1903.10318*.

Yang Liu and Mirella Lapata. 2019. Text summarization with pretrained encoders. *arXiv preprint arXiv:1908.08345*.

Chirantana Mallick, Ajit Kumar Das, Madhurima Dutta, Asit Kumar Das, and Apurba Sarkar. 2019. Graph-based text summarization using modified textrank. In *Soft computing in data analytics*, pages 137–146. Springer.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Rupert G Miller Jr. 1997. *Beyond ANOVA: basics of applied statistics*. CRC press.

Mudasir Mohd, Rafiya Jan, and Muzaffar Shah. 2020. Text document summarization using word embedding. *Expert Systems with Applications*, 143:112958.

Khanh Nguyen and Hal Daumé III. 2019. Global voices: Crossing borders in automatic news summarization. *arXiv preprint arXiv:1910.00421*.

Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab. Previous number = SIDL-WP-1999-0120.

Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE transactions on Signal Processing*, 45(11):2673–2681.

Abigail See, Peter J Liu, and Christopher D Manning. 2017. Get to the point: Summarization with pointer-generator networks. *arXiv preprint arXiv:1704.04368*.

Krithi Shetty and Jagadish S Kallimani. 2017. Automatic extractive text summarization using k-means clustering. In *2017 International Conference on Electrical, Electronics, Communication, Computer, and Optimization Techniques (ICEECCOT)*, pages 1–9. IEEE.

George R Terrell and David W Scott. 1992. Variable kernel density estimation. *The Annals of Statistics*, pages 1236–1265.

Daniel Varab and Natalie Schluter. 2020. Danewsroom: A large-scale danish summarisation dataset. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6731–6739.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Xingxing Zhang, Mirella Lapata, Furu Wei, and Ming Zhou. 2018. Neural latent extractive document summarization. *arXiv preprint arXiv:1808.07187*.

Ming Zhong, Pengfei Liu, Yiran Chen, Danqing Wang, Xipeng Qiu, and Xuanjing Huang. 2020. Extractive summarization as text matching. *arXiv preprint arXiv:2004.08795*.

Qingyu Zhou, Nan Yang, Furu Wei, Shaohan Huang, Ming Zhou, and Tiejun Zhao. 2018. Neural document summarization by jointly learning to score and select sentences. *arXiv preprint arXiv:1807.02305*.

A Appendix

Summary	Article
Gold Summary	Denmark's new innovation network, Danish Healthtech, is now making matchmaking available to Danish companies. These companies will thus be able to be connected with Danish educational institutions to use knowledge within, among other things, cyber security and health technology. Aalborg University (AAU) is currently working on projects within diagnosing cardiovascular disease and measuring hyperactivity in the central nervous system, and it is with projects such as these that Danish companies can become partners. In addition to AAU, the network's partners include DTU, AU, SDU and KU.
Candidate Sentences	[1] About the Innovation Network Danish Healthtech The Innovation Network Danish Healthtech is based on a unique partnership in healthcare technology that ensures easy access for small and large companies to relevant knowledge institutions across Denmark. [2] As a national innovation network, the ambition for Danish Healthtech is to contribute to solving the societal challenges within health through increased knowledge about the use and development of technological solutions. [3] The network is based on close collaboration between the four regional clusters for health technology - LifeScience Innovation (LSI), The network builds on experiences from two previous innovation networks: MedTech Innovation's strong medical technology position and the strong position within welfare technology, which the Welfare Tech Innovation Network for Health and Welfare Technology has built up. [4] Together with the Maersk Mc-Kinney Møller Institute at the University of Southern Denmark, companies have the opportunity to collaborate with researchers who focus on designing and developing new innovative services using visual data analysis. [5] Based on the experiences from previous networks, Danish Healthtech will continue to facilitate matchmaking. [6] Matchmaking and knowledge sharing are paramount The Innovation Network's most important role is to act as a matchmaker between companies and knowledge institutions. [7] The purpose is to present research in the field of telehealth, to discuss opportunities and barriers between research and development of technology products. [8] Danish Healthtech offers companies to get closer to the knowledge institutions in the broad field of health technology. [9] Danish Healthtech meets the demand with new conferences, workshops and more collaborative projects in 2019 and 2020. [10] Medtech Innovation Consortium, Welfare Tech and Copenhagen Healthtech Cluster - and at the same time a merger of the two innovation networks, MedTech Innovation (MTI) and the Welfare Tech Innovation Network for Health and Welfare Technology (ISV). [11] Danish Healthtech has several meetings in the pipeline between clinicians, researchers, companies, municipalities and citizens in the form of idea generation workshops. [12] topic that has been of great interest among the MedTech Innovation network's players in 2018 with a sold-out conference and two completed innovation projects. [13] As a company, together with Danish Healthtech, you have the same opportunities for knowledge collaboration as Elos Medtech. [14] The knowledge of companies, hospitals and other actors must be strengthened through seminars, workshops and collaborative projects. [15] An innovation project supported by MedTech Innovation in 2018 with the participation of, among others, Elos Medtech and DTU on surfaces on dental implants has led to an expanded collaboration that now also involves Aarhus University, which has thus received 10 million kroner from Grand Solutions (Innovationsfonden).

Table A.1: The gold summary and the first 15 candidate sentences of the article ranking by LexRank. This is an English version translated from the original Danish version via Google Translate.