COMP550 Final Project: Comparing Extractive Text Summarization Methods Across Languages

Kira Noël
260846383
McGill University

Nakiya Noorbhai
260844714
McGill University

Luca Garnier-Landurie
260930881
McGill University

Abstract

Text summarization is an important field in NLP, but key research efforts have, up to now, almost exclusively focused on English. As we become an increasingly global society, it is crucial that other languages also be able to benefit from these advances. Unfortunately, state-of-the-art text summarization techniques require large datasets, which are currently unavailable in most languages (Oufaida et al., 2015). To address this problem, we explore whether extractive text summarization techniques traditionally used for English work well for other languages. We find that the tested methods are primarily language-independent, although further optimization is possible on a per-language basis.

1 Introduction

With the abundance of textual data that we interact with today, being able to easily access and interpret this data is of the utmost importance. Moreover, given the increasing availability of non-English textual resources, it is equally as important to have tools that can not only make text more accessible and readable, but that can do so for information presented in any language, not just English (Oufaida et al., 2015). Given that most NLP research focuses on English, our paper examines the degree to which current, popular NLP approaches used for English can be applied to other languages. More specifically, we examine whether extractive text summarization models offer a potential solution to making non-English text more readable and accessible. Summary generation is a valuable tool for transforming long and inaccessible text into a more manageable, concise and comprehensible form. (Moratanch and Chitrakala, 2017). We hypothesize that, given the language-independent nature of many extractive text summarization models, current text summarization approaches designed and used for English-language research can work equally well for other, less-researched languages. (Kaity and Balakrishnan, 2020).

2 Related work

2.1 Text Summarization

Research into automatic text summarization has been ongoing since the early 1950s (195). There has been a significant amount of research that focuses on implementing novel approaches for text summarization (Moratanch and Chitrakala, 2017). Major discoveries include clustering based algorithms including LSA, graph based algorithms such as TextRank and, more recently, various neural and metaheuristic approaches (Verma et al., 2019).

Automatic text summarization is classified into abstractive and extractive text summarization(Verma et al., 2019). Abstractive text summarization uses complex models to generate an original text summary, whereas extractive text summarization methods builds a summary by concatenating statistically or linguistically significant sentences taken from the input text (Moratanch and Chitrakala, 2017). While abstractive text summarization generates more-human like summaries, abstractive models must be trained on large amounts of language-specific data and require significant computational resources to run (Oufaida et al., 2015).

2.2 Extractive Text Summarization

One of the main advantages of extractive text-summarization is that, unlike with abstractive text-summarization, many popular, unsupervised extractive text summarization models do not require significant amounts of language-specific data (Mihalcea, 2005). Moreover, because of their unsupervised nature, these models are language independent which gives them the potential to perform well

on relatively understudied languages such as Russian or Turkish (Mihalcea, 2005). Given that one of the major challenges in NLP is the difficulty of obtaining non-English language resources and data, this makes extractive text summarizers particularly practical. (Oufaida et al., 2015).

Because of this, a significant portion of the work in the field of automatic text summarization revolves around the discovery, implementation and optimization of various extractive text summarization models that can be used to summarize text in non-english languages. There is significant research on implementing novel extractive models as well as research examining how pre-existing algorithms can be modified to improve their performance on non-english languages (Moratanch and Chitrakala, 2017). Examples of work in this field include introducing new LSA based text summarization methods to better summarize Turkish text, examining the potential of using mRMR discriminant analysis to better summarize Arabic and French text, and even data augmentation approaches that supplement pre-existing German-language data with additional synthetically produced data to improve the performance of a text summarizer (Ozsoy et al., 2010; Oufaida et al., 2015; Parida and Motlicek, 2019).

Our paper takes a more practical approach. That is, instead of generating more language specific data, coming up with novel summarization approaches, or optimizing existing approaches to text-summarization, we simply explore how well we can summarize news articles in other languages given only currently available algorithms and language-specific resources.

Our methods are heavily inspired by the work of Verma et al. (2019) and Mihalcea (2005). The Verma et al. (2019) paper examines the effectiveness of Luhn, TextRank and eleven other extractive text summarization models on both English and Hindi news articles. Similarly, Mihalcea (2005) examines how graph structure variations impact the quality of summaries produced by a TextRank model run on English and Brazilian Portuguese news article data. Like with our own experiments, these two papers use the ROUGE metric to evaluate how well existing algorithms summarize news articles written in non-English languages. We extend on these studies by using data from six different languages instead of looking at just two. More importantly, while the focus of these studies is to find an optimal model or to optimize existing models for a specific language, we are not focused on optimizing or selecting the best models. Instead, our paper serves as a checkpoint to try and understand our current ability to summarize non-English text.

3 Method

3.1 Datasets

We use two datasets of pairs of full-text news articles and their human-created summaries: one which contains only English texts and summaries from CNN and Daily Mail (See et al., 2021), and the other which contains records for five other languages from similar news organizations (Scialom et al., 2020). Together, the two datasets contain approximately 300k records for each of English, French, German, Spanish, and Turkish, and 30k for Russian. They are each divided into development and test sets. Due to the low computing power available and that extractive text summarization models do not require training data, we limit the development and test sets to 5k entries each, except for Russian, whose test set only has 1.5k entries because of the limited available data.

3.2 Models

We test two approaches to extractive text summarization: Luhn and TextRank. Both algorithms are implemented by popular Python libraries *gensim* and *sumy*. However, since these implementations non-optionally apply language-specific stopwords and stemmers to the input, we implement our own versions.

3.2.1 Luhn

The main assumption of Luhn's method is that frequent terms are important, with the exception of stopwords. In this implementation, the score is computed by looking at the ratio of important (top percent of most frequent words) to unimportant words in a sentence (Fig. 1). This implementation closely follows that of Vashisht. This method is often implemented with stemming, but due to lack of availability of stemmers for less-studied languages, we do not use one. Stopword lists may not be available for less-studied languages, but they are easier to create, so we optionally remove provided stopwords.

3.2.2 TextRank

The TextRank approach instead assumes that if there are many sentences that are similar, they are

$$\text{score} = \frac{\text{\# important words}^2}{\text{\# unimportant words}}$$

Figure 1: Equation used for scoring the sentences with Luhn's method.

important. To implement this, the words in the sentences are vectorized using term frequency inverse document frequency (TF-IDF) and a similarity matrix is computed. The scores are then obtained by applying the PageRank algorithm from the Python package *networkx* to a graph computed from the similarity matrix. The similarity matrix is generated in two ways: with the unigram overlap and with cosine similarity (implemented with *Word2Vec* from the Python package *gensim*). We test these independently on each dataset. The implementation is adapted from that of Gupta (2020).

3.3 Metric

The quality of text summaries is difficult to measure automatically. Ideally, we would have people read both the machine-generated summaries and the articles to judge the quality of the summaries. Given the time constraint of this project and that we do not speak all of the tested languages, this option was not feasible. Instead, we make the assumption that good summaries will be similar. We compare the machine-generated summaries to the human-written ones and say that the more similar they are, the better the machine-generated one.

We use the ROUGE-N F_1 score as the metric to compare the performance of the different summarization approaches across each language. More specifically, we use the mean ROUGE-1 score, which determines similarity between two texts through the number of unigrams in common. A higher score means that the texts are more similar. Note that we always ensure that the machine-generated summary has the same number of sentences as the human reference summary to prevent over- or under-scoring.

3.4 Experiment design

First, we compute a baseline by generating summaries through random sampling of sentences from the text without replacement. The summaries are scored against the references according to section 3.3. This gives us a baseline score that can be used to determine the efficacy of the other summarization methods.

Next, summaries are generated using each of the extractive text summarization models discussed above and record the corresponding ROUGE F_1 scores. In each case, the input text is lowered and punctuation is removed before processing. For each of the models, we try with and without removing stopwords (language-specific stopwords obtained from Python library *nltk*). For the Luhn approach, we test different percentages $(1, 2, 4, 8, 16, 32, 64, 100)$ of the most frequent words to consider meaningful; these percentages are referred to as *thresholds*.

We run these experiments twice: once with the dev set and again with the test set to confirm that our findings are not limited to just one part of the dataset.

4 Results

It is important to note two facts before discussing the results. First, ROUGE scores can only be compared meaningfully within a language because different languages have characteristics that would affect the scoring (Moratanch and Chitrakala, 2017; Verma et al., 2019). Second, as this is *extractive* text summarization and the reference summaries are *abstractive*, it is expected for the ROUGE scores to be low. We must consider the scores with respect to the random baseline in each language.

Also, we obtain but omit the ROUGE-2 scores, as they follow the same trends as the ROUGE-1 scores in all cases.

	Base	Luhn		TR Uni		TR Cos	
	-	S	N	S	N	S	N
en	.20	**.26**	.23	.24	.22	.20	.20
fr	.14	**.29**	.26	.21	.19	.14	.14
de	.16	**.40**	.31	.22	.17	.16	.16
ru	.05	**.07**	**.07**	**.07**	**.07**	.05	.05
es	.13	**.17**	**.17**	.16	.15	.13	.13
tu	.08	**.18**	.17	.12	.11	.08	.08

Table 1: ROUGE-1 F_1 scores for the dev sets of each language. *Uni* indicates unigram, and *cos* indicates cosine similarity. *S* indicates that stopwords are removed and *N* indicates that stopwords were not removed. The scores for Luhn are taken with an optimized threshold. The best score for each language is bolded.

4.1 Comparing Summarization Methods

For all languages except Russian, text summarization with Luhn's method offers the best performance on both dev and test sets (Fig. 1, Fig. 2). TextRank using unigram overlap as the sentence

	Base	Luhn		TR Uni		TR Cos	
	-	S	N	S	N	S	N
en	.25	**.33**	.30	.30	.27	.24	.24
fr	.14	**.19**	.18	.17	.17	.14	.13
de	.16	**.22**	.18	.16	.14	.12	.12
ru	.04	.06	**.07**	.06	.06	.04	.04
es	.15	**.19**	**.19**	.18	.17	.15	.14
tu	.15	**.26**	.25	.19	.19	.15	.15

Table 2: ROUGE-1 F_1 scores for the test sets of each language. *Uni* indicates unigram, and *cos* indicates cosine similarity. *S* indicates that stopwords are removed and *N* indicates that stopwords were not removed. The scores for Luhn are taken with an optimized threshold. The best score for each language is bolded.

ranking metric also performs well and consistently improves on the baseline. However, TextRank using cosine similarity as the ranking metric gives poor results; the score is consistently as bad or worse than the random baseline.

4.2 Optimization for Luhn's Method

In Figure 2, we see that most languages have optimal performance using Luhn's method with a threshold of 32% (Fig. 2, 3). German and Spanish achieve slightly better performance with smaller thresholds when stopwords are removed. However, for Spanish without stopwords removed, the difference in performance between thresholds of 4% and 32% is slight (0.0002 and 0.004 for the dev and test sets, respectively). The difference is more pronounced for the German dataset, as seen in Figure 4; although only the test set is shown, the same pattern is observed in the dev set. Russian is also an exception: Luhn's method performs best here for very small thresholds.

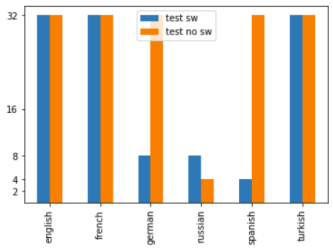

Figure 2: Optimal lower thresholds on the test set for each language for text summarization with Luhn's method. The same trends are observed for the dev set, so these results are omitted. Blue indicates that stopwords are removed and orange means that stopwords are not removed.

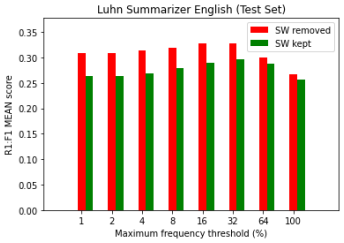

Figure 3: ROUGE-1 F_1 scores for the English test set using Luhn's method. *SW removed* indicates that the stopwords are removed from the text and *SW kept* indicates that they were not.

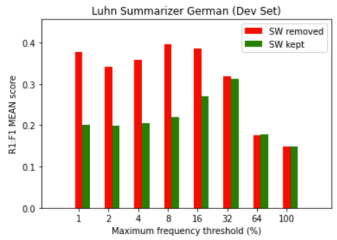

Figure 4: ROUGE-1 F_1 scores for the German test set using Luhn's method. *SW removed* indicates that the stopwords are removed from the text and *SW kept* indicates that they were not.

4.3 Optimization with Stopwords

Removing stopwords either improves or does not affect the ROUGE score for all languages and all models on both dev and test sets, except Russian for which the score decreases slightly with Luhn's method on the test set (Fig. 2). We see a particularly large increase with the German dataset, particularly with the dev set. The improvement is not always large, but it is consistent.

4.4 Russian as an Outlier

The model performance on the Russian dataset does not follow the same trends as the rest of the data. Its ROUGE-1 scores are very low (< 0.08), and while text summarization with Luhn's method and TextRank using unigrams both improve over the baseline, the improvement is very slight (0.02-0.03). This is the one language for which removing stopwords actually decreases performance in one instance (Luhn's method without removing stopwords outperforms Luhn's method with stopwords removed for the test set). Last, Luhn's method is optimal for much smaller lower thresholds than for any of the other languages (Fig. 2).

5 Discussion and conclusion

5.1 Overall Model Performance

As shown by results 4.1, 4.2 and 4.3, the extractive text summarization methods examined perform well for many languages. More specifically, we saw the same model performance trends across all languages, except for Russian. With Luhn's model, all non-Russian languages achieved optimal or, in the case of German and Spanish, near-optimal ROUGE scores at a 32% threshold. Further, removing stop words improved performance across most models and did not decrease performance in any except Russian. Finally all examined languages performed poorly with the cosine TextRank model. While this poor performance is likely due to there not being enough data to properly form appropriate word vectors to yield meaningful cosine similarity values, it can still be seen as another indication that the models performance does not significantly vary across languages. That is, as model performance follows similar trends for the majority of languages, it is likely that the Luhn and TextRank extractive text summarization models are, for the most part, language-independent and perform similarly when used with both English and many other languages.

5.2 Outliers

Russian did not follow the same performance trends as any of the other examined languages. It consistently performed poorly across all models. This is potentially due to the types of Russian-language news articles found in the datset. Our dataset contains significantly fewer Russian records. Due to this lack of record variety, it may be that many of the news articles are written in a similar style that makes it difficult for the models to create appropriate summaries.

That said, while the datset may contribute to the low ROUGE scores, it is more likely that Russian performed poorly because it cannot appropriately be evaluated with the ROUGE score metric. We achieved optimal ROUGE scores well above 0.15 for all languages except for Russian whose optimal ROUGE score was consistently 0.07. Moreover, we were unable to find any studies that used the ROUGE score metric to evaluate Russian model performance. For these reasons, the unusual and poor performance of Russian might be due to its incompatibility with our evaluation metric rather than its inability to be effectively summarized by extractive summarization models.

5.3 Limitations

While our experiments produced interesting results, it is worth noting that our study has many limitations. The main limitations being our evaluation methods. We use the ROUGE metric to compare our model generated summaries to the human-written summaries from our dataset. As mentioned above, not all languages may be compatible with the ROUGE metric and so it may not be the most robust way to evaluate performance. Moreover, we use the human-generated summaries from our dataset as gold-standards. This is also a significant limitation as there is not only one way to write a summary. A model-generated summary may be equally as good as a syntactically- or semantically-different model-generated summary. Our experiment does not take this into account. For these reasons, a good extension of our experiment may be use other evaluation criteria such as "readability", "cohesiveness", and "conciseness" to evaluate the generated summaries instead of just using rouge score (Moratanch and Chitrakala, 2017). As for other ways to extend our research, it may also be helpful to examine other languages that are more linguistically-dissimilar to English to ensure that our results hold more generally.

5.4 Conclusions

In conclusion, we find that the majority of our results are in line with our hypothesis. That is, we find that the implemented versions of Luhn and TextRank extractive text summarization models perform similarly for both English, and the majority of tested non-English languages. However, we also acknowledge that further experiments with more robust evaluation metrics and a broader scope of examined languages are required to gather further evidence for the language-independence of extractive text summarization methods and to fully understand why we receive inconsistent results for Russian.

6 Statement of contributions

Kira wrote the majority of the code and worked on the method and results sections of the report. Nakiya did background research, prepared the bulk of the report, and analysed the results. Luca ran some of the experiments and compiled the plots. We all contributed to designing the experiment and editing the report.

References

Mehul Gupta. 2020. Text summarization using textrank in nlp.

Mohammed Kaity and Vimala Balakrishnan. 2020. Sentiment lexicons and non-english languages: a survey. *Knowledge and Information Systems*, 62.

Rada Mihalcea. 2005. Language independent extractive summarization. In *Proceedings of the ACL 2005 on Interactive Poster and Demonstration Sessions*, ACLdemo '05, page 49–52, USA. Association for Computational Linguistics.

N. Moratanch and S. Chitrakala. 2017. A survey on extractive text summarization. In *2017 International Conference on Computer, Communication and Signal Processing (ICCCSP)*, pages 1–6.

Houda Oufaida, Philippe Blache, and Omar Nouali. 2015. Using distributed word representations and mrmr discriminant analysis for multilingual text summarization. pages 51–63.

Makbule Ozsoy, Ilyas Cicekli, and Ferda Alpaslan. 2010. Text summarization of turkish texts using latent semantic analysis. In *Proceedings of the 23rd international conference on computational linguistics (Coling 2010)*, pages 869–876.

Shantipriya Parida and Petr Motlicek. 2019. Abstract text summarization: A low resource challenge. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5994–5998, Hong Kong, China. Association for Computational Linguistics.

Thomas Scialom, Paul-Alexis Dray, Sylvain Lamprier, Benjamin Piwowarski, and Jacopo Staiano. 2020. Mlsum: The multilingual summarization corpus. *arXiv preprint arXiv:2004.14900*. Data retrieved from Hugging Face, https://huggingface.co/datasets/mlsum.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2021. Dataset: cnn_dailymail / 3.0.0. Data retrieved from Hugging Face, https://huggingface.co/datasets/cnn_dailymail.

Ashutosh Vashisht. Luhn's heuristic method for text summarization. Accessed 2021.

Pradeepika Verma, Sukomal Pal, and Hari Om. 2019. A comparative analysis on hindi and english extractive text summarization. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(3).