Methods for increasing cohesion in automatically extracted summaries of Swedish news articles

Elsa Andersson
Dept. of Computer and Info. Science
Linköping University
Linköping, Sweden
elsa.andersson@liu.se

Arne Jönsson
Dept. of Computer and Info. Science
Linköping University
Linköping, Sweden
arne.jonsson@liu.se

Abstract

Summaries created by extractive models trained on Swedish texts often lack cohesion, which affects the readability and overall quality of the summary. This paper explores and implements methods at the data-processing stage aimed at improving cohesion of generated summaries. The methods are based around Sentence-BERT for creating sentence embeddings that can be used to rank sentences in a text in terms of if it should be included in the extractive summary or not. Three models are trained using different methods and evaluated using ROUGE and BERTScore for measuring content coverage and Coh-Metrix for measuring cohesion. The results of the evaluation suggest that the methods can indeed be used to create more cohesive summaries, although content coverage was reduced, which gives rise to the potential for future exploration.

1 Introduction

Monsen and Rennes (2022) compared extractive and abstractive summaries of Swedish news articles in terms of readability and text quality through an online survey and concluded that extractive summaries were perceived to be more fluent and adequate, and were often preferred over its abstractive counterpart. This could be an argument for choosing extractive over abstractive summarization if the purpose is to create coherent summaries.

While there are many benefits of extractive summarization and many techniques to use for the task, it comes with its own set of challenges and flaws that are particularly hard to clear up. One category of challenges are the errors made due to the difficulties of handling references and coreferences (Sukthanker et al., 2020). These errors still occur in extracted summaries due to the requirement of hard-to-acquire background knowledge (Mitkov et al., 2001). Creating coherent and cohesive summaries are two of the most challenging tasks in automatic text summarization (Gambhir and Gupta, 2017), yet considerably important when aiming for the production of high-quality summaries. One step on the way is tackling one text genre at a time and then combining the knowledge to ultimately have the ability to manage a much wider set of different Swedish text types. While the goal is not to create genre-specific methods that are only applicable on one text type, news articles have several benefits (length, structure, content, etc.) in terms of simplicity over other text types when working with summarization tasks. Therefore, news articles have become and remain a standard choice within the field of NLP on, not only, the Swedish language.

The research presented in this paper consists of altering the data-processing stage of training an extractive summarization BERT model that can improve the readability of automatically generated summaries on Swedish news articles. Specifically, the purpose is to improve cohesion while maintaining good content coverage of the summaries.

2 Methods to improve summary generation

The process for an automatic text summarizer (ATS) to go from input data (single- or multi-document) to generating what will further be referred to as a candidate summary can be split into three stages: pre-processing, processing and post-processing (El-Kassas et al., 2021). In this paper we will only present results based on pre-processing and processing. Post-processing methods treat the generated candidate summary in order to further increase quality.

Reformatting data for the purpose of extractive summarization is a key step when creating an automatic extractive summarizer. It is the task of ranking sentences in a document, which can be

done using a variation of methods. With the development and breakthroughs in deep-learning networks, such as BERT, it has quickly become the most popular category of method to use for the extractive summarization task. However, prior to this shift, many other types of methods were used (and still are), such as graph-based (e.g. (Mallick et al., 2019)), semantic-based (e.g. (Mohamed and Oussalah, 2019)) and statistical-based (e.g. (Afsharizadeh et al., 2018)). TransformerSum[1] offers a method that determines which sentences in an article should be included in the summary and labels them 0 (should not be included) or 1 (should be included). This is determined by maximizing the ROUGE score between a generated extractive summary and the manually created summary for each article. Note that the manually created summary, or the gold standard, is not an extractive summary but an abstractive one.

2.1 Creating sentence embeddings

While the BERT architecture has led to incredible performances in certain NLP tasks, there are several limitations when it comes to other regression tasks that use sentence pairs, such as semantic textual similarity (STS), due to the vast amount of sentence combinations that are required. Sentence-BERT (SBERT) addresses this issue by modifying the architecture into a siamese and triplet structure which adds a pooling layer to the output from the BERT-model, and sequentially creates sentence embeddings that are semantically meaningful and comparable using cosine similarity. Furthermore, sentence embeddings created with SBERT can be used in the pre-processing stage of extractive summarization when reformatting the data. However, as the data is in Swedish, a method of creating cross-lingual sentence embeddings has to be adapted.

2.2 Extending sentence embeddings to learn novel languages

Many models that compute sentence embeddings perform badly when creating sentence representations for non-English languages (Reimers and Gurevych, 2020). Luckily, Reimers and Gurevych (2020) has presented a method that uses knowledge distillation to extend sentence embeddings from one model that was created on a specific language (typically English) to a novel language.

[1] https://transformersum.com

Novel in this sense means in a language that the model has not yet been trained on. The method is to train a "student" model to work on a novel language by mapping sentence embeddings created by a well-performing "teacher" model (trained on English for example) to parallel sentences in the novel language. The teacher model provides sentence embeddings for English sentences in a set of data that contains English sentences and the same sentences translated to the novel language. Those embeddings are subsequently mapped to the same sentences in the new language by the student model, creating a vector space model that is often much better than if the model was to be trained on only the novel language.

3 Evaluation metrics

The quality of a summary can be evaluated from two standpoints; content coverage and text cohesion. The rate of content coverage corresponds to how much, or little, of the important content of the original text is retained in the summary. Text cohesion is tightly tied to the readability of the summary and requires semantic linking through references between sentences to be intact, otherwise leading to confusion, nonsensical statements or altered meaning (Kaspersson et al., 2012). In order to measure content coverage and cohesion of the summaries generated, three different metrics will be used. For content coverage, ROUGE (Lin, 2004) and BERTScore (Zhang et al., 2019), will be used, while Coh-Metrix (Zhang et al., 2019) will be used to measure cohesion.

4 Data

The data was produced by Monsen and Jönsson (2021) for training and evaluating abstractive summarization models, and then further filtered and adapted by Monsen and Rennes (2022) for training and evaluating both abstractive and extractive models. It consists of 349,935 article-summary pairs from the largest Swedish morning newspaper Dagens Nyheter (DN) between the years 2000-2020. The summaries for each article correspond to the associated hand-written preamble of the article.

After filtering, the dataset was split into 339,935 pairs for training, 9,000 for testing and 1,000 for validation. The average article length in the training set is 476 words or 30.3 sentences and the average summary length is 33 words or 2.5 sentences.

	ROUGE		
	ROUGE-max	Distiluse	MiniLM
ROUGE-1	**0.2838**	0.2673	0.2662
ROUGE-2	**0.0922**	0.0840	0.0817
ROUGE-L	**0.1746**	0.1606	0.1604
ROUGE-Lsum	**0.1745**	0.1607	0.1604

Table 1: Average ROUGE scores for each model. The highest score for each metric is marked in bold

5 Methods

The method suggested by TransformerSum requires the data to be structured into source and target files, where the source files consisted of the articles and the target files the manually created summaries to be used as the gold standard. The model that is trained on data to maximize ROUGE scores will be referred to as **ROUGE-max**.

To use the sentence transformer method, we used a pre-trained model to compute cosine-similarities for each sentence in the article with the summary and scored them that way. The model that is trained on data processed this way will be referred to as **Distiluse.**

SentenceTransformers offers an extensive evaluation of pre-trained models in semantic search and sentence embedding tasks on more than 1 billion training pairs. We used paraphrase-multilingual-MiniLM-L12-v2 as the student model and paraphrase-MiniLM-L12-v2 as the teacher model[2]. The model that is trained on data processed this way will be referred to as **MiniLM.**

After processing the data in the three different ways described above, it was further processed and three models were trained using TransformerSum which allows for further pre-processing (creating features and tokenizing data) that prepares the data to be inserted into the model for training. First, features were created and the processed dataset was tokenized by utilizing the pre-trained BERT model for Swedish tasks (Malmsten et al., 2020), which was trained on a variety of sources such as books, news articles and the Swedish Wikipedia, and has a vocabulary size of approximately 50,000 words. Finally, the models were trained and fine-tuned, using 1,000 warm-up steps and a total of 60,000 training steps. Batch-size was set to 16 and the number of epochs set to 3. This process was done three times and resulted in the three models ROUGE-max, Distiluse and MiniLM.

[2] https://www.sbert.net/_static/html/models_en_sentence_embeddings.html

	BERTScore		
	ROUGE-max	Distiluse	MiniLM
mean	**0.1530**	0.1393	0.1367
std	**0.0941**	0.0893	0.0886

Table 2: BERTScore for each model. The highest score for each metric is marked in bold.

6 Results

ROUGE-max performed the best out of the three models in all of the ROUGE metrics, as can be seen in Table 1. Distiluse and MiniLM had nearly identical scores with the highest difference being 2.78%, while ROUGE-max scored at least 5.99% higher in all metrics compared to both Distiluse and MiniLM.

As Table 2 illustrates, ROUGE-max scored the highest on the BERTScore metric, with a 9.37% difference in mean BERTScore and 5.24% difference in standard deviation compared to the second highest (Distiluse). Distiluse had slightly higher scores compared to MiniLM; 1.88% higher in mean BERTScore and 0.79% in standard deviation.

As Table 3 illustrates, Distiluse and MiniLM had the highest scores in all Coh-metrix metrics. Distiluse scored the highest in adjacent and global arguments and nouns, and adjacent stems. MiniLM scored the highest on adjacent and global anaphors and global stems.

As is illustrated in Table 4, Distiluse and MiniLM performed better than ROUGE-max on all connectives metrics with a minimum difference of 5.69% on all metrics apart from temporal connectives, where the difference was 0.71%. MiniLM scored better than Distiluse only on the additive connectives metric. However, the scores of Distiluse and MiniLM differed by at the most 1.17% apart from the temporal connectives metric, which differed by 3.96%.

As Table 5 shows, ROUGE-max performed only slightly higher compared to the other models when measuring LSA scores, and MiniLM scored slightly higher than the other models on givenness. The difference in scores from the adjacent average metric shows a difference of less than one percent: 0.41% between ROUGE-max and Distiluse and 0.52% between Distiluse and MiniLM. The standard deviation of adjacent LSA scores between the ROUGE-max and Distiluse and Distiluse and MiniLM differed by 4.86% and 4.72% respectively. Finally, the difference when comparing the models on givenness score was 0.99% between MiniLM

	Referential cohesion					
	ROUGE-max		Distiluse		MiniLM	
	adjacent	*global*	*adjacent*	*global*	*adjacent*	*global*
anaphors	0.4407	0.2202	0.4513	0.2276	**0.4694**	**0.2421**
arguments	0.3899	0.3866	**0.4045**	**0.4004**	0.3921	0.3984
nouns	0.4548	0.4566	**0.5087**	**0.5040**	0.4902	0.4867
stems	0.3413	0.1707	**0.3605**	0.1798	0.3540	**0.1810**

Table 3: Average referential cohesion scores for each model. The highest score for each metric is marked in bold. Each referential cohesion metric was measured on an adjacent level and a global level. The adjacent level considers two sentences in a pair, while the global level considers every possible sentence pair in the summary.

	Connectives		
	ROUGE-max	Distiluse	MiniLM
CNCCaus	0.9822	**1.0497**	1.0397
CNCADC	1.1466	**1.3118**	1.3073
CNCAdd	3.8869	4.3018	**4.3524**
CNCAll	9.2537	**10.0559**	9.9602
CNCTemp	3.2379	**3.3924**	3.2608

Table 4: Average incidence scores of connectives for each model. The highest score for each metric is marked in bold. CNCCaus = Causal Connectives; CNCADC = Adversative/contrastive Connectives; CNCAdd = Additive Connectives; CNCAll = All Connectives; CNCTemp = Temporal Connectives.

	LSA		
	ROUGE-max	Distiluse	MiniLM
adjacent avg	**0.5389**	0.5367	0.5339
adjacent std	**0.0843**	0.0803	0.0766
givenness	0.4635	0.4580	**0.4681**

Table 5: Average LSA scores for each model. The highest score for each metric is marked in bold. LSA (Latent Semantic analysis) is merely computed on an adjacent (sentence pairs) and single sentence level as a global level entails a paragraph-paragraph analysis, and each summary consists of only one paragraph.

and ROUGE-max and 1.19% between Distiluse and ROUGE-max.

7 Discussion

MiniLM and Distiluse outperformed ROUGE-max by a substantial amount in nearly all cohesion metrics, apart from LSA which indicated no or minimal difference between the models. This implies that overall, cohesion increased. However, ROUGE-max outperformed both Distiluse and MiniLM in regard to content coverage, implying that content coverage suffered. When comparing MiniLM and Distiluse in terms of cohesion, there is no apparent trend that points to one being superior to the other; they performed equally (connectives and LSA) or varied in equal ways (referential cohesion). The same conclusion applies to content coverage, as neither Distiluse nor MiniLM outperformed the other in either of the content coverage metrics.

The results point to the implication that sentence transformers and the transfer of knowledge through sentence embeddings are useful methods when trying to train models with the aim of producing more cohesive summaries.

While Distiluse and MiniLM perform equally in the evaluation measures, there are other factors that can contribute to the choice of method. One such factor is the amount of time it takes to train each model. ROUGE-max took considerably less time to train in contrast to the other two models, where some training steps took up to 20 hours each. This also means that Distiluse and MiniLM required more processing power. Other factors could include difficulty of implementation, where for example ROUGE-max required much less altering of the provided framework from TransformerSum compared to the other two models. Availability/access might also influence the choice, where some pretrained models are easier to find than others. Therefore, the factors have to be weighed in order to make a decision based on the resources that can be spent on training. Perhaps, performance in terms of summary quality is worth sacrificing for the convenience of less difficulty, overall time spent and processing load.

Furthermore, post-processing for extractive summarization can be done using several methods, such as the reordering of sentences in the summary, solving errors with anaphoric references and co-references by replacing pronouns with their antecedents and replacing temporal expressions with dates, to mention a few (El-Kassas et al., 2021)(Gupta and Lehal, 2010). This could further increase the cohesion of summaries and does not require heavy processing, meaning it can be done on more limited hardware.

References

Mahsa Afsharizadeh, Hossein Ebrahimpour-Komleh, and Ayoub Bagheri. 2018. Query-oriented text summarization using sentence extraction technique. In *2018 4th international conference on web research (ICWR)*, pages 128–132. 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.

Mahak Gambhir and Vishal Gupta. 2017. Recent automatic text summarization techniques: a survey. *Artificial Intelligence Review*, 47(1):1–66.

Vishal Gupta and Gurpreet Singh Lehal. 2010. A survey of text summarization extractive techniques. *Journal of emerging technologies in web intelligence*, 2(3):258–268.

Thomas Kaspersson, Christian Smith, Henrik Danielsson, and Arne Jönsson. 2012. This also affects the context-errors in extraction based summaries. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 173–178.

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

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.

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden–making a swedish bert. *arXiv preprint arXiv:2007.01658*.

Ruslan Mitkov, Branimir Boguraev, and Shalom Lappin. 2001. Introduction to the special issue on computational anaphora resolution. *Computational Linguistics*, 27(4):473–477.

Muhidin Mohamed and Mourad Oussalah. 2019. Srl-esa-textsum: A text summarization approach based on semantic role labeling and explicit semantic analysis. *Information Processing & Management*, 56(4):1356–1372.

Julius Monsen and Arne Jönsson. 2021. A method for building non-english corpora for abstractive text summarization. In *Proceedings of CLARIN Annual Conference 2021*.

Julius Monsen and Evelina Rennes. 2022. Perceived text quality and readability in extractive and abstractive summaries. In *Proceedings of the 13th international conference on Language Resources and Evaluation (LREC), Marseille, France*.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.

Rhea Sukthanker, Soujanya Poria, Erik Cambria, and Ramkumar Thirunavukarasu. 2020. Anaphora and coreference resolution: A review. *Information Fusion*, 59:139–162.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.