THExtended: Transformer-based Highlights Extraction for News Summarization

Luca Agnese
DAUIN
Torino, Italy
l.agnese@studenti.polito.it

Alessio Paone
DAUIN
Torino, Italy
s306030@studenti.polito.it

Flavio Spuri
DAUIN
Torino, Italy
s303657@studenti.polito.it

Luca Zilli
DAUIN
Torino, Italy
s303356@studenti.polito.it

***Abstract*—The summarization process has gained significant attention in the field of Natural Language Processing (NLP) over the past decade, finding widespread application across various domains such as news summarization and extraction of salient portions from scientific papers, including highlights. The general objective is to identify and select the most N important sentences from a given text corpus, thereby creating a summary that effectively captures the underlying meaning of the document.**

In today's digital age, the volume of news and information generated on a daily basis has become overwhelming. With the advent of social media and online platforms, the accessibility and speed of news dissemination have increased exponentially. However, this deluge of information poses a challenge for individuals seeking to stay informed without getting overwhelmed.

This work focuses on utilizing a Neural Network-based method for extractive summarization. In particular, it leverages the work of THExt, a Transformer-based pipeline to extract summaries from scientific papers, adapting it to the news domain. Furthermore, an exploration of the trade-off between syntactic and semantic evaluation is carried out. For the purpose of fine-tuning and evaluating our system, we use the CNN/DailyMail dataset, which is a commonly used resource in research related to the news domain. *Source Code*: github.com/Raffix-14/THExtended/

***Index Terms*—News summarization, CNN/DailyMail, Extractive summarization, Transfomers**

I. Introduction

Text summarization stands out as a highly promising task within the field of Natural Language Processing (NLP). Its aim is to condense a textual document into a shorter set of highlights while retaining salient information of the original corpus. We specifically focus on the extractive summarization task, since it is considered the more computationally efficient and grammatically correct approach by the literature [1]. This task addresses the growing challenge of effectively managing the ever-increasing volume of news content [2]. This increase in the number of news articles can be attributed to the exponential expansion of official news channels and the adoption of social media platforms; by developing our work, we aim to offer a viable solution to this growing issue by enhancing the news search capabilities for end-users that could enable them to retrieve information more efficiently.

Our research builds upon THExt [3], a framework that employs a Transformer-based architecture [4] as the sentence-level encoder for highlights extraction. Specifically, the problem is formulated as a regression task that feeds as input to the regression head the contextualized attention-aware embeddings of the Transformer. The values predicted by the regressor are then compared to the scores obtained on the real summary, i.e. the human-annotated collection of highlights.

A first challenge when moving from the scientific paper domain, considered in THExt, to our domain of news articles, is the definition of a suitable context. Indeed, we lose the structure intrinsic to the papers, which are divided into standardized sections, moving to textual data lacking any predefined organization. This made the context extraction method from the original work inapplicable to our case. Thus, a suitable context definition was sought through an exploratory phase, aiming to identify the section that most effectively captures the overarching meaning of the article.

Expanding on THExt, we also introduce an analysis of the trade-off between using a syntactic and a semantic score to perform the extractive summarization task.

To recap our contributions:

- First, we adapted the THExt framework to the news domain. To do so, we also needed a new way to define a meaningful context, as we lose the intrinsic structure present in the scientific paper.
- Second, we analyzed the trade-off between using a semantic score and a similarity score by introducing a novel modified scoring mechanism.

II. Related works

The adoption of deep learning techniques for text summarization has seen a significant rise, as they often result in state-of-the-art performances across various benchmark datasets. Historically, the most effective strategies for this specific task have relied on GRUs or LSTMs networks. However, recently, Transformer-based approaches have been progressively replacing these traditional methods with superior performance.

Some first remarkable models to consider are SummaRuNNER [5], a GRU-based sequence model that approaches the task as a binary classification, and the successive work of NeuSum [6]. This latter is an end-to-end hierarchical document encoder and sentence extractor network based on GRUs that jointly learns to score and select sentences, thus moving toward continuous labels. This last method successfully exceeds the previous state-of-the-art performances on CNN/DailyMail among other datasets.

Despite the impressive achievements, the highlighted models still suffer from a few drawbacks. Specifically, they require a large corpus for pre-training and are generally difficult to interpret.

Acknowledged the remarkable outcomes of deep learning methods using RNNs, it is important to note their inherent limitation of forgetting long-range text dependencies. Transformer-based approaches effectively address this by leveraging the multi-head attention mechanism, which provides a broader attention span across the input sequence.

One of the first notable methods that employ an efficient Transformer-based approach is MatchSum [1]. It is a novel framework that leverages a semantic text-matching approach to extractive summarization. More specifically, a Siamese-BERT architecture is proposed to compute the similarity between the source document and the candidate summary trained with a margin-based triplet loss.

Finally, we mention THExt [3], the paper upon which our methodology is largely based. The overall architecture employs a Transformer-based approach, in which the encodings of the sentences are computed and then fed to a regression head. Being tailored to scientific papers, this approach takes into account the embedding generated from both the single sentences and a context, i.e. the most significant section of the paper. This helps the model accurately assess the importance of the target sentence, and is one of the main contributions brought by the work.

III. Methodology

A. Dataset

In the conduct of our research, we utilized the CNN/DailyMail (v3) dataset [7]. This latest iteration is designed to support both extractive and abstractive summarization. It incorporates a collection of articles, each paired with a multi-sentence summary. specifically, the dataset contains 286,817 training pairs, 13,368 validation pairs, and 11,487 test pairs. A selection of general statistics regarding the dataset can be found in Table I.

TABLE I
General information before preprocessing

Section	Avg num. words	Avg num. sentences
Article	766	30
Summary	53	4

Following the methodology outlined in the original THExt work, our ultimate goal is to extract a number K of highlights, where K in a hyper-parameter spanning in $\{3,4,5\}$. Thus, our dataset preprocessing begins with filtering out the samples with summaries that are constituted by less than 3 or more than 5 sentences. We note that, since we will simply select the K sentences that the model assesses to be most likely highlights, it would suffice to consider articles with at least 3 ground-truth highlights. However, we deem that considering articles that present a ground-truth summary of more than 5 highlights would unfairly increase the overall performance. As depicted in the figure [?], we observed that some articles start with a few metadata that contains information such as the newspaper name, the date, and the article location. Given that these additional pieces of information do not contribute to our intended usage and could potentially introduce confounding elements for the model, we further preprocessed the dataset so as to filter them out from most of the articles.

The final step of dataset preprocessing encompassed the more challenging task of splitting each article into individual sentences. In order to obtain sentences containing a sound semantic meaning, we couldn't employ a basic rule-based split. Instead, we leveraged a language pipeline provided in the spaCy library, specifically *en_core_web_lg* [8].

B. Metrics

In this section, we delve into the specific evaluation metrics that we utilized in our study. These metrics, widely accepted in the field, offer quantitative insights into the performance of our model. They allow us to measure the quality of the extracted highlights, both in terms of syntactic similarity and in terms of retained semantic meaning.

- **Rouge-N.** Rouge-N is a set of metrics commonly used for evaluating automatic summarization. Specifically, it measures the overlap of N-grams between the system-generated and the reference highlights.
- **Rouge-L.** Rouge-L is another metric commonly used in text summarization. It computes the longest common subsequence between the highlights.
- **Semantic score.** To determine the semantic similarity between a generated and a reference highlight, we consider the cosine similarity between their contextualized embedding vector. To balance performance and computational cost, we use *all-MiniLM-L6-v2* [9], a Transformer-based sentence-level contextualized embedder that obtains competitive performances although greatly reducing the number of parameters.

For all metrics within the Rouge-family, we consider both precision, recall, and the F1 measure. Among these, F1-score will be our main metric of reference, because of its invariance with respect to the order in which the highlight pairs are given.

C. Model architecture

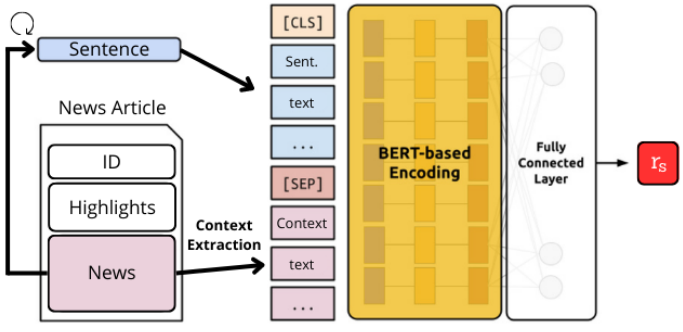

Fig. 1. Overview of the model architecture

Our network architecture consists of two primary components, as illustrated in Figure 1.

The first component is an encoder which transforms the sentences into a meaningful representation. For this purpose, we utilized BERT [4], a multi-layer bidirectional Transformer, further details of which can be found in the original study [10]. The key benefit of this model is its capability of considering long-range dependencies, achieved via the multi-head attention mechanism.

Following the encoding of sentences, these representations are passed onto our second component: a Fully Connected Layer. This layer performs a regression task, eventually yielding the final relevance scores associated to the sentences.

D. News domain adaptation

Considering the aforementioned unstructured nature of our text corpus, we conducted an analysis of how to redefine a suitable context. In particular, we tried to identify which part of an article encapsulates most of the overarching meaning, making it the optimal one to add in addition to the candidate sentence.

Our methodology consists of analyzing three evenly spaced segments from the articles in our corpus. For each segment, we generate an embedding using the sentence encoder Longformer [11], as standard BERT encoding presents a maximum capability of 512 tokens, insufficient for our particular needs. Each embedded section is then compared to the entire article's embedding via cosine similarity. The most suitable section is thus taken as the one resulting in the highest similarity with regard to the whole article. The final results of the analysis can be observed in Figure 2. As suggested from the data, the first segment captures the majority of the overall meaning by a considerable margin. Consequently, this segment has been chosen as our context.

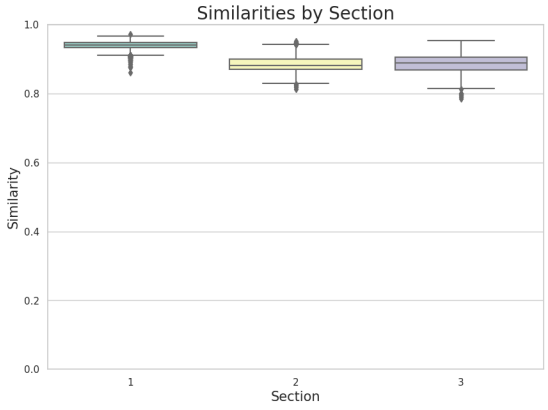

Fig. 2. Average relevance of various segments of an article

In accordance with one of the main innovations to the methodology presented in THExt, the primary component of our network is not limited to encoding single sentences. Rather, the recently defined context is fed into the encoder as well, with the hope of injecting meaningful context that augments the model's capacity to better represent the sentences.

E. Combining Semantic and Syntactic Scores

Following the original research, the training process aims at teaching the network how to predict the syntactic similarity of a sentence with regard to a reference highlight. Specifically, this is accomplished by considering the max F1 score of the Rouge-2 metric between the sentence and the reference highlights as the final label of the network.

To expand on that, we analyzed how partially injecting information about the semantic similarity would impact the network performances. In this context, we considered as the training label a convex combination of the F1 score of the Rouge-2 metric and the semantic similarity as defined in III-B, controlled by a parameter α

$$l_s = \max_{h \in H} \alpha R_2(s,h) + (1-\alpha) S(s,h) \tag{1}$$

where:

$$\begin{aligned} l_s &= \text{train label for sentence}_s \\ H &= \text{set of reference highlights} \\ R_2(s,h) &= \text{F1 score of the Rouge-2 metric} \\ S(s,h) &= \text{semantic similarity} \end{aligned} \tag{2}$$

IV. Experiments

In the following section, we will present a comprehensive overview of the experiments conducted to evaluate the methodologies previously described.

A. Implementation details

As detailed in III-C, the first component of our architecture uses BERT to encode our text. In particular, among the various BERT versions available to us, we implemented the *base uncased* pre-trained variant of the model [12], via the HuggingFace library [13]. This version is pre-trained on a general text corpus and is thus not specifically tailored to the news domain. A version of BERT with further pre-train steps in this specific domain would yield improved performance, but to the best of our knowledge no such version is currently publicly available, nor do we have the computational capacity to perform such steps independently.

We developed our project on Google Colab Pro© [14]. This environment provided us with V100 NVIDIA GPUs and 15 GB of RAM as training hardware. As a result, each training required an approximate training time of 4 hours.

The hyperparameters used for the training of our models are the following:

- **Optimizer.** AdamW [15], with the default parameters as specified in the PyTorch implementation.
- **Learning Rate.** we selected $3e^{-5}$, a choice commonly seen in the training of BERT-based architectures.
- **Batch Size.** 64, the maximum permissible given our computational resources. However, we also implemented *gradient accumulation* with a step equal to 2, effectively achieving a batch size of 128.
- **Epochs.** we performed 2 *fine-tuning* epochs for each model

- **Warmup Steps.** given our limitations in training capacity, it is set to 0.

B. News Domain Adaptation

To examine the performance when adapting the work done in THExt to the news domain, we conducted a series of experiments, the results of which can be found in the second section of Table II. Specifically, we trained our model using two distinct scores. Following the definitions introduced in III-E, we consider a purely syntactic score (R_2) and a purely semantic one (S).

Those two experiments are denoted in the Table as $\alpha = 1$ and $\alpha = 0$, coherently with the general formula introduced in III-B. To evaluate their performances, we separately considered their results in terms of the F1-score of the Rouge-1, Rouge-2, and Rouge-L metrics, and in terms of the Semantic Score. As one might predict, employing purely syntactic or semantic scores during the training phase leads to better performance at test time on syntactic or semantic scores, respectively. In particular, when changing α from 0 to 1, we observe an increase in syntactic scores by approximately 1 percentage point (i.e. an overall increase of around 6%) and a decrease in the semantic score by approximately 2 percentage points (i.e. an overall decrease of around 3%)

Due to our inability to perform any pre-training steps, directly comparing our results with other methodologies proposed for the same dataset would be of limited value. To still gain some insight into the performance of our models, we introduce two control experiments. In the first, named *Lead*, we simply select as our k highlights the first k sentences of the articles. In the second, named *Oracle*, we assume perfect knowledge of the reference summary during the evaluation process, providing us with an upper bound for our performance. The results of these experiments can be seen in the first section of Table II. From these, it is apparent that while there remains considerable room for improvement in relation to the upper bound, our methodology significantly outperforms the naive implementation. We also note that, since the performance of *Oracle* is well below the theoretical maximum of 1, this indicates the dataset's highlights are, for a considerable portion, abstractive.

C. Combining Semantic and Syntactic Scores

For our second set of experiments, we examined how different combinations of the syntactic and semantic scores at the training phase would impact the performance on said scores separately. To do so, we run additional experiments setting $\alpha \in \{0.25, 0.5, 0.75\}$, accordingly to the methodology introduced in III-E. Our ultimate aim was to determine which combinations yield the optimal trade-off between performances concerning both syntactic and semantic scores.

The outcomes of these experiments can be seen in the third section of Table II. The general behavior observed can be summarized as follows: upon progressively decreasing α from 1 (purely syntactic) towards 0, we initially notice that the performance in terms of syntactic scores, namely the F1-scores of the Rouge-1, Rouge-2, and Rouge-L metrics, stays approximately the same, even presenting a marginal increase. Meanwhile, we observe some greater improvement in the Semantic score. However, once α drops below a certain threshold —specifically, for values strictly smaller than 0.5 in our experiments—, we only observe minor improvements in the Semantic Score, while the syntactic scores start degrading.

The best-performing value for α in terms of our metric of interest, the F1-score of the Rouge-2 metric, seems to be at 0.5. Overall, the best balance between the syntactic and semantic evaluations appears to fall within the range of $\alpha \in [0.5, 0.75]$. To illustrate it, we attempt to depict the Pareto optimality boundary in Figure 3. However, it's worth noting that to conduct a thorough analysis of the Pareto optimality, we would need to train for many additional values of α, as our limited samples only allow us to infer the overall shape of the boundary.

TABLE II
RESULTS FOR VARYING VALUES OF α

Setup	R1	R2	RL	SemScore
Lead	0.2720	0.1418	0.2456	0.5682
Oracle	0.4769	0.3306	0.4517	0.4353
$\alpha = 1$	0.3286	0.1852	0.3013	0.6164
$\alpha = 0$	0.3201	0.1754	0.2915	0.6368
$\alpha = 0.25$	0.3258	0.1802	0.2974	0.6364
$\alpha = 0.5$	0.3332	0.1868	0.3045	0.6351
$\alpha = 0.75$	0.3328	0.1866	0.3045	0.6281

Note: $\alpha = 1$ denotes full reliance on a syntactic score, while $\alpha = 0$ on a semantic score. **R1 R2**, and **R3** refer to the F1-scores of the Rouge-1, Rouge-2, and Rouge-L respectively

V. CONCLUSION

Our study successfully adapts the approach introduced in THExt for extractive summarization to the domain of news. In particular, we fine-tuned and measured the performances on the well-known CNN/Daily Mail benchmark dataset.

Moreover, we evaluate the effectiveness of combining syntactic and semantic scores as a target function. The results show that introducing partial semantic information increases the performance of the model even when taking a syntactic score as the main metric of reference. This is most likely due to the nature of the considered dataset since, although it is sponsored as suitable for extractive summarization, contains mainly abstractive highlights.

Future works. Assuming appropriate resources, as future work one could consider testing our methodology on a wider range of datasets. Moreover, it would be possible to enhance our model's performance by further training it with a larger sample size. Exploring alternative BERT-based encoders could also be investigated. Additionally, we deem a new version of BERT, pre-trained on news, could be a great contribution to literature and it would probably give a great boost to the performance.

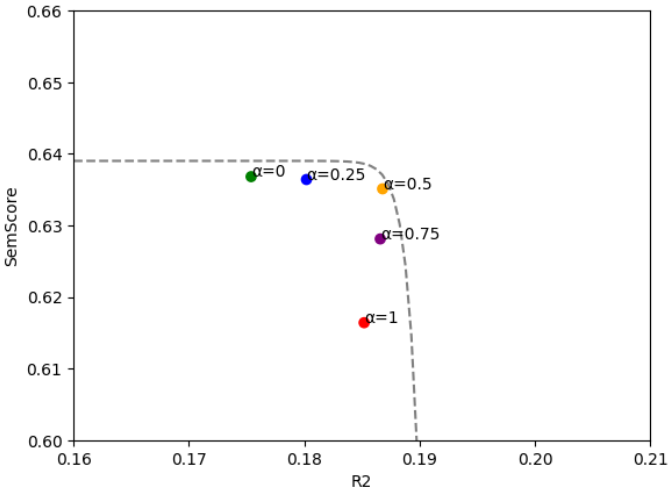

Fig. 3. Orientative Pareto optimality curve

REFERENCES

[1] M. Zhong, P. Liu, Y. Chen, D. Wang, X. Qiu, and X. Huang, "Extractive summarization as text matching," *arXiv preprint arXiv:2004.08795*, 2020.

[2] V. Narayanan, V. Barash, J. Kelly, B. Kollanyi, L.-M. Neudert, and P. N. Howard, "Polarization, partisanship and junk news consumption over social media in the us," *arXiv preprint arXiv:1803.01845*, 2018.

[3] M. La Quatra and L. Cagliero, "Transformer-based highlights extraction from scientific papers," *Knowledge-Based Systems*, vol. 252, p. 109382, 2022.

[4] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[5] R. Nallapati, F. Zhai, and B. Zhou, "Summarunner: A recurrent neural network based sequence model for extractive summarization of documents," in *Proceedings of the AAAI conference on artificial intelligence*, vol. 31, no. 1, 2017.

[6] Q. Zhou, N. Yang, F. Wei, S. Huang, M. Zhou, and T. Zhao, "Neural document summarization by jointly learning to score and select sentences," *arXiv preprint arXiv:1807.02305*, 2018.

[7] K. M. Hermann, T. Kociský, E. Grefenstette, L. Espeholt, W. Kay, M. Suleyman, and P. Blunsom, "Teaching machines to read and comprehend," in *NIPS*, 2015, pp. 1693–1701. [Online]. Available: http://papers.nips.cc/paper/5945-teaching-machines-to-read-and-comprehend

[8] M. Honnibal and I. Montani, "spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing," 2017, to appear.

[9] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, and M. Zhou, "Minilm: Deep self-attention distillation for task-agnostic compression of pre-trained transformers," *CoRR*, vol. abs/2002.10957, 2020. [Online]. Available: https://arxiv.org/abs/2002.10957

[10] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[11] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[12] "('bert-base-uncased') from huggingface," https://huggingface.co/bert-base-uncased, accessed: 2023-09-19.

[13] "Huggingface library," https://huggingface.co/, accessed: 2023-09-19.

[14] "Google colaboratory," https://colab.research.google.com/?hl=it, accessed: 2023-09-14.

[15] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," 2019.