Extractive Text Summarization using Text Segmentation

Mattia Bertè* **Simone Giordani*** **Omar G. Younis***
{mattia.berte, simone.giordani2, omargallal.younis}@studio.unibo.it

1 Executive summary

Single-document summarization is the task of generating a shorter version of a document while retaining its most important content. There are two fundamental approaches to text summarization: extractive and abstractive. The former extracts words and word phrases from the original text to create a summary. The latter learns an internal language representation to generate more human-like summaries, paraphrasing the intent of the original text. Here we compare two methods for the extractive text summarization: a graph based method called PACSUM (Zheng and Lapata, 2019), and a method of our proposal, that consist on performing text segmentation, retreiving the argumentative units in the document, and filter them based on their class. We compare the results using the ROUGE metric (Lin, 2004), very popular in the text summarization research area.

2 Methods

2.1 Graph Based Method - PACSUM

A very popular algorithm for extractive single-document summarization is TEXTRANK (Mihalcea and Tarau, 2004). It represents document sentences as nodes in a graph with undirected edges whose weights are computed based on sentence similarity. In order to decide which sentence to include in the summary, a node's centrality is often measured using graph-based ranking algorithms such as PAGERANK (Brin and Page, 1998). In (Zheng and Lapata, 2019), they improved the centrality measure in two ways: Firstly, to better capture sentential meaning and compute sentence similarity, they employ BERT (Devlin et al., 2018), a neural representation learning model which has obtained state-of-the-art results on various natural language processing tasks. Secondly, they adopted a directed graph, with different weights to the forward and backward edges, since the contribution induced by two nodes' connection to their respective centrality can be in many cases unequal. In fact sometimes a phrase would not make much sense on its own, without the support of some preceding sentence, and so the weights of the two edges should be different. They called this method PACSUM (Position-Augmented Centrality based Summarization).

Given a document D composed by the sentences $\{s_1, s_2, ..., s_n\}$, an encoding $\{v_1, v_2, ..., v_n\}$ of each sentence in the document is obtained using a pretrained model (BERT in the paper). Then pair-wise dot product is computed to obtain an unnormalized similarity matrix $\bar{e}$. The similarity score is then normalized using equation 2 to remove the effect of absolute values by emphasizing the relative contribution of different similarity scores. This is particularly important for the adopted sentence representations which in some cases might assign very high values to all possible sentence pairs. Hyper-parameter $\beta \in [0, 1]$ controls the threshold below which the similarity score is set to 0. The next step is to compute the centrality of each sentence, using equation 3, where λ_1 and λ_2 are different weights for forward and backward-looking directed edges. Finally the summary is computed by extracting the k sentences with the highest centrality score, where k is a predefined number.

$$\bar{e}_{ij} = v_i^\top v_j \tag{1}$$

$$e_{ij} = \max(0, \bar{e}_{ij} - [\min \bar{e} + \beta(\max \bar{e} - \min \bar{e})]) \tag{2}$$

$$\textbf{centrality}(s_i) = \lambda_1 \sum_{j<i} e_{ij} + \lambda_2 \sum_{j>i} e_{ij} \tag{3}$$

2.2 Text Segmentation - DETR

The goal of the Text Segmentation task is to identify argumentative elements from a document, localizing their position in the document, and classifying them based on the following classes:

- **Lead** - an introduction that begins with a statistic, a quotation, a description, or some other device to grab the reader's attention and point toward the thesis
- **Position** - an opinion or conclusion on the main question
- **Claim** - a claim that supports the position
- **Counterclaim** - a claim that refutes another claim or gives an opposing reason to the position
- **Rebuttal** - a claim that refutes a counterclaim
- **Evidence** - ideas or examples that support claims, counterclaims, or rebuttals.
- **Concluding Statement** - a concluding statement that restates the claims

Applying this categorization to Text Summarization is quite straightforward: lots of part of an argumentative essay, despite being useful in the full document to explain better some concept and, most importantly, to prove that the thesis is supported by evidence, in a summary turns out to be useless or redundant. Examples of those classes are the *Counterclaim* or the *Rebuttal*, which sole purpose is to make the speech more critical, without adding information. So we try to summarize those kind of texts by extracting only those classes that brings the most information. As in out previous project we used a model inspired by DETR (Carion et al., 2020), a state-of-the-art model for object detection based on transformers, using as backbone a LongFormer (Beltagy et al., 2020), useful to precess very large documents. The model has been trained on a dataset that contains more than 15k argumentative essays written by U.S students in grades 6-12, annotated by expert raters.

2.3 ROUGE evaluation metric

ROUGE score, introduced in (Lin, 2004), has developed to become the standard evaluation measure for summarization tasks, in the research community. ROUGE stands for *Recall-Oriented Understudy for Gisting Evaluation*. Although it is *recalled-oriented* as its name suggest, in essence, it considers both Recall and Precision between candidate (model-generated or predicted) and reference (golden-annotated or target) summaries. It is evaluated by looking for corresponfing n-grams, tokens of n consecutive words, between reference and candidate summaries. It evaluates the *recall*, the proportion of n-grams in the reference summary captured by the candidate summary, and the *precision* the proportion of n-grams suggested by the candidate summary, that actually appear in the reference summary, and then is evaluate the *f-1 score* as the armonic mean $F1 = 2(P + R)/PR$.

ROUGE-1 and ROUGE-2 consider respectively single words and bigrams, ROUGE-L instead considers the *Longest Common Subsequence* (LCS) words. LCS refers to word tokens that are in sequence, but not necessarily consecutive. Together, ROUGE-1, ROUGE-2 and ROUGE-L give a good representation of how well the model-generated summaries represent the golden-annotated summaries.

3 Experimental setup

We conducted our experiments on a small dataset of 10 argumentative essays taken from the dataset used to train DETR. We produced one golden-annotated summary for each of them, by extracting the 3 sentences that, in our opinion, best express the meaning of the complete text.

In our implementation of PACSUM, we used as sentence encoder a pretrained model taken from the huggingface model hub [1], and from all the possible choices, we selected `all-mpnet-base-v2`. This is an all-round model, derived from MPNet (Song et al., 2020), tuned for many use-cases. It is trained on a large and diverse dataset of over 1 billion training pairs using a self-supervised contrastive learning objective. Given a sentence from the pair, the model should predict which out of a set of randomly sampled other sentences, was actually paired with it in the dataset. We then conducted an hyperparameter search to find the best values of β and λ_1 (as in the paper $\lambda_2 = 1 - \lambda_1$ to reduce the hyperparameter space), using as objective score the mean value of the ROUGE f-1 scores. The best values are reported in table 1. As expected turns out that λ_2 should be grater than λ_1, because it is more likely that a sentence

[1] https://www.sbert.net/docs/pretrained_models.html

refers to a previously expressed concept, and therefore if that sentence were chosen for the summary, it would risk making little sense without the context offered by the previous sentences.

Instead for our DETR summary extractor we looked for the best set of classes to extract, trying all possible combinations of the 7 classes. As expected, the best classes are those that introduce the arguments and express the final considerations, as they contain all the necessary concepts that the author of the essay wants to express. The other classes are then filtered because their content is not generic enough to summarize the concepts.

The results of the experiments using both methods separately and together, in a cascade fashion with DETR as first filter and then PACSUM as sencences selector, are reported in table 2

PACSUM		DETR Classes
β	0.22	Lead
λ_1	0.44	Position
λ_2	0.56	Concluding Statement

Table 1: Best hyperparameters used

	ROUGE-1	ROUGE-2	ROUGE-L	Mean
PACSUM				
recall	0.617	0.503	0.597	**0.572**
precision	0.717	0.586	0.693	**0.666**
f-1 score	0.658	0.536	0.636	**0.610**
DETR				
recall	0.773	0.629	0.743	**0.715**
precision	0.540	0.400	0.520	**0.486**
f-1 score	0.626	0.477	0.602	**0.568**
DETR + PACSUM				
recall	0.593	0.453	0.563	**0.536**
precision	0.678	0.529	0.647	**0.618**
f-1 score	0.627	0.483	0.597	**0.569**

Table 2: Comparison of results using both methods separately and together

4 Discussion

As shown in table 2, PACSUM performs slightly better that out method. One explanation is that the sentence encoder is a much more well trained model compared to our DETR model, which didn't perform very well even in his task of text segmentation, obtaining an f1-score of just 0.29, and this leads also in poorer performances in this task. The most noticeable error in the predictions is on the start and end of sentences. That's because the model was trained without constraints on the punctuation, and so didn't learn well how to recognize the start and the end of a sentence. For instance, as shown in the example reported in the Apprendix, DETR misses and the last words "*greatest achivements.*" in the concluding statement.

Nevertheless some considerations in the comparison between the two algorithms can be drawn. Fist of all our method report a much higher recall compared to PACSUM. This means that many of the salient parts of the summaries are actually present in the classes selected by our algorithm. On the other hand the precision is much lower, meaning that the summaries that it produces contains some useless sentences. The reason for this can be found both in the poor training of our network, but also in the method by which we created the dataset. By selecting only 3 sentences for each summary we have implicitly defined a constraint, while probably the length of the summary should be variable according to the length of the text. Perhaps a larger and different dataset would reward the feature of our algorithm of not having a predefined number of sentences to extract.

Lastly the performance of the cascade method return to be similar to the ones with only PACSUM. One possible explanation is that, as said before, the sentences extracted by DETR may be incomplete, and so the encoding of those sentences may have a lower similarity with the others. But if the segmentation was correct, then the results would probably be better. This statement is supported by the example reported in the Appendix, where we corrected the prediction of DETR by adding the missing words "*greatest achivements.*" in the concluding statement. It can be seen that PACSUM changed its choice on the last sentence, agreeing to keep that concluding statement, and this leads to the best f-1 score across all the methods in this particular example, as shown in table 3. We believe that this pattern can be repeated in other examples as well, with a better trained DETR model.

References

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems*, 30(1):107–117. Proceedings of the Seventh International World Wide Web Conference.

Nicolas Carion, Francisco Massa, Gabriel Synnaeve, Nicolas Usunier, Alexander Kirillov, and Sergey Zagoruyko. 2020. End-to-end object detection with transformers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain, July. Association for Computational Linguistics.

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, Barcelona, Spain, July. Association for Computational Linguistics.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. *CoRR*, abs/2004.09297.

Hao Zheng and Mirella Lapata. 2019. Sentence centrality revisited for unsupervised summarization. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6236–6247, Florence, Italy, July. Association for Computational Linguistics.

Appendix - Example of summarization

Full Text

the driverless cars are a bad idea. in theroy, they sound wonderful. but in reality they are a safty hazard. to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u.s.the driver has to stay aware of the other things around them. even thought you still have to keep your hands on the wheel because of the touch sensors. the article states, in fact, none of the cars developed so far are completely driverless. this statment shows there is still room for human error in these driverless cars.even though human error is still involved, these cars still can't go through construction areas or around wrecks but they still alert you when you come across it.the prompt states, but all are designed when the road requires human skills, such as navigating through work zones and around acciedents. this means the human driver must remain alert and be ready to take over when the situation requires. people are still needed to drive these cars, but we still have to do the hard part the old fashion way.even with accident warning, these cars are still not legal in some states like california and nevada. for example, as a result it is illegal even to test computer driven cars. california, nevada, florida, and the district of columbia have led the country in allowing limited use of semi autonomous cars. they don't think it's safe enough to be driven in highly populated areas.these driverless cars are a hazard. even though they could be helpful in the future. in some way they seem to undertested amd people don't want to trust them. manly because you would still have to pay attention to the road, still have to drive through construction and wrecks, and are technically still not fully legalised in some states like california, nevada and florida. with a bit more field testing these driverless cars could be one of our greatest achivements. but they won't be totally ready until 2020.

Reference

the driverless cars are a bad idea.to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u.s.people are still needed to drive these cars, but we still have to do the hard part the old fashion way.

PACSUM

to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u. these driverless cars are a hazard. this statment shows there is still room for human error in these driverless cars.

DETR

the driverless cars are a bad idea. in theroy, they sound wonderful. but in reality they are a safty hazard. to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u.s.the way they seem to undertested amd people don't want to trust them. manly because you would still have to pay attention to the road, still have to drive through construction and wrecks, and are technically still not fully legalised in some states like california, nevada and florida. with a bit more field testing these driverless cars could be one of our

DETR + PACSUM

to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u. the driverless cars are a bad idea. manly because you would still have to pay attention to the road, still have to drive through construction and wrecks, and are technically still not fully legalised in some states like california, nevada and florida.

DETR + correction + PACSUM

to start with the driver needs to stay aware of the car and traffic, they cant go through construction, and are still are not legal in some states around the u. the driverless cars are a bad idea. with a bit more field testing these driverless cars could be one of our greatest achivements.

	PACSUM	DETR	DETR + PACSUM	DETR + correction + PACSUM
recall	0.624	**0.766**	0.695	0.671
precision	0.729	0.455	0.636	**0.739**
f-1 score	0.673	0.570	0.664	**0.703**

Table 3: Mean ROUGE scores of the example