Extractive Summarization for Explainable Sentiment Analysis using Transformers

Luca Bacco[a,b,c], Andrea Cimino[b], Felice Dell'Orletta[b] and Mario Merone[a]

[a] *Università Campus Bio-Medico di Roma, Unit of Computer Systems and Bioinformatics, Dep. of Engineering*
[b] *Istituto di Linguistica Computazionale "Antonio Zampolli" (ILC–CNR), ItaliaNLP Lab*
[c] *Webmonks s.r.l.*

Abstract

In recent years, the paradigm of eXplainable Artificial Intelligence (XAI) systems has gained wide research interest and beyond. The Natural Language Processing (NLP) community is also approaching this new way of understanding AI applications: building a suite of models that provide an explanation for the decision, without affecting performance. This is certainly not an easy task, considering the wide use of very poorly interpretable models such as Transformers, which in recent years are found to be almost ubiquitous in the NLP literature because of the great strides they have allowed. Here we propose two different methodologies to exploit the performance of these models in a task of sentiment analysis and, in the meantime, to generate a summary that serves as an explanation of the decision taken by the system. To compare the classification performance of the two methodologies, we used the IMDB dataset while, to assess the explainability performance, we annotated some samples of this dataset to retrieve human extractive summaries, benchmarking them with the summaries generated by the systems.

1. Introduction

As more and more content is shared by people on the web, the use of automated ***S**entiment **A**nalysis (SA)* tools has become increasingly present. Just think of solutions for monitoring public opinion on social media, or for drawing feedbacks from products and/or services reviews, to understand what consumers like and don't. However, today's systems often lack transparency, as they cannot provide an interpretation of their reasoning. In recent years, this has been a well-known problem in the scientific community. In fact, the contribution that ***A**rtificial **I**ntelligence (AI)* algorithms are making in shaping tomorrow's society is constantly growing. Given the high performance that today's models can achieve, their application is spanning an increasingly large landscape of fields. This is motivating a rapid paradigm shift in the use of these technologies. Indeed, we are moving from a paradigm in which *AI* models are required to deliver the highest possible performance, to one in which such systems are required to provide information about taken decisions that is interpretable by humans. We are referring to the ***eX**plainable **A**rtificial **I**ntelligence (XAI)* paradigm. As stated by *DARPA*'s *XAI* program launched in 2017, the main goal of *XAI* is to create a suite of models that provide an explanation without affecting performance [1, 2, 3]. That is, to pass from the concept of black-box models,

l.bacco@unicampus.it (L. Bacco); andrea.cimino@ilc.cnr.it (A. Cimino); felice.dellorletta@ilc.cnr.it (F. Dell'Orletta); m.merone@unicampus.it (M. Merone)

© 2021 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).
CEUR Workshop Proceedings (CEUR-WS.org)

in which it is hard (or even impossible) to get any sort of explanation from them, to white-box ones, in which the model also provides results that are understandable by the final users, or at least by the experts in the application domain [4]. This may lead systems of the near future to address the needs of government organizations and the users who use them, such as the right to explanation, which can raise the reliability of users in the system, and the right to decision rejection, especially in applications where a human-the-loop approach is expected (*Articles 13-15, 22 of the EU GDPR*). In addition, the ***N**atural **L**anguage **P**rocessing (NLP)* community is beginning to approach to this new paradigm [5]. However, the task of explaining *NLP* systems is certainly not an easy one, in a context where models based on deep neural networks, usually referred to as the least explicable models of machine learning, take the lead. In fact, since the *Transformer* architecture was introduced by Vaswani et al. [6] (Sec. 2.2), the *NLP* research has made great strides. In an effort to investigate the behaviour of these models and provide some sort of human-understandable interpretation, the weights of the attention mechanism inherent in these structures have often been taken into account (Sec. 2.4). In this work, we proposed and compared two Transformer-based models to perform tasks of sentiment analysis, while retrieving an explanation of the models' decisions through a summary built by extracting the sentences of the document that are the most informative for the task in hand. That is, we exploited the *extractive (single document) summarization* paradigm (Sec. 2.1). In particular, for one of the two models, we made use of the attention weights of the Transformer model to get insights on the most relevant sentences. To do so, we exploited a hierarchical configuration (Sec. 2.3). To evaluate our proposed models, we tested their effectiveness on a binary sentiment classification task. However, the underlying structures may be easily adapted for any document classification task. To evaluate the classification performance, we used the IMDB movie reviews dataset [7] while, to assess the explainability performance, we annotated some samples of the dataset to retrieve human extractive summaries from the training and test sets, and then assessed the overlap between these and the models' ones. The annotation phase was necessary since there are not so many works in literature dealing with the explainability side of sentiment analysis models, and most of them deal with it by highlighting the most relevant words based on some importance scores. The main contributions of this work may be resumed as: collecting a new dataset for the evaluation of extractive summaries as an explanation of a sentiment analysis task; exploring the use of attention weights of a hierarchical transformer architecture as a base to achieve extractive summaries as an explanation of a document classification task; comparing two different models, both based on transformer architectures, in terms of the performance in both the classification and explanation tasks.

2. Related Works

2.1. Automatic Text Summarization

The Automatic Text Summarization (ATS) topic is gaining more and more interest in research, not only in the academic but also in the industrial field. This is due to the increasingly large amount of textual data on the various archives of the Internet. It is not difficult to imagine the value it may have to automatically summarize scientific papers, to give an example close to our world. In addition, such an approach could be beneficial for analyzing, clinical documents

(usually, kinds of documents that are very long), social media opinions, product reviews, etc. From these points of view, it becomes even more obvious how it would be worthy to automatize a summarization process if you think about how much a Manual Text Summarization (MTS) may cost, in terms of both time and human effort. Not least, the ATS may be used as an explanation of a model decision. For example, as in this work, a summary may explain the classification given by the model of some documents. However, ATS is not a monolithic topic of research, but it may be seen as spread in many sub-fields where researchers are putting their efforts in. Following the nomenclature in [8], we may distinguish the first and most important differences between ATS techniques presented in the literature. First of all, ATS systems may be classified by the size of their input. We may have a system which target is to shorten a single document given in input (*SDS*, Single Document Summarization) or to compress the important pieces of information from a set of multiple documents (*MDS*, Multi-Document Summarization). However, the MDS paradigm is not suitable for the case at hand, where we were interested in achieving an interpretation (the summary) on the classification of a single document. Systems may also be divided by the nature of the summary. Some methods are defined as *extractive*, because they build summaries by extracting the most important sentences from the document. Others are called *abstractive*, because they aim to generate a summary made by new (generated) sentences. Even if the abstractive paradigm can theoretically solve issues like redundancy and information lost, because of the task complexity the research efforts focused more on the extractive kind. A third way is the *hybrid* one, that may be seen as a trade-off between the two paradigms. By the way, since our models focus on extracting sentences from the original document, it falls within the extractive paradigm. Furthermore, we could define our models also as generic, because they aim to give a general sense of the document to justify the decision, and deep learning-based (because, of course, Transformers are deep neural networks models). For an in-depth analysis of the nomenclature of the summarization systems, we suggest the reader to refer to [8].

2.2. Transformers vs. RNNs

Since modeling the contextual content in documents is a key point to success in many NLP tasks such as document classification, Recurrent Neural Networks [9, 10] (RNNs) had an increasingly growing trend in the computational linguistic community. At least, prior to the advent of Transformers models [6]. In fact, even with the Bidirectional variant [11] (Bi-RNNs), such networks are intrinsically sequential. This means that their use is limited to restricted corpora because of their expensive computational cost. Furthermore, due to two phenomena during the training phase, named exploding gradient and vanishing gradient [12], the dependency of the text of a sequence is limited to not so long context. However, their variation with Long-Short Term Memory [13] and Gated Recurrent Unit [14] cells (LSTMs and GRUs) helped to partially overcome this issue. In fact, just a few years ago, it was not so surprising to see these networks applied to complex NLP tasks, such as Language Modeling (LM) [15, 16]. However, since 2017, the interest of the NLP community in this kind of networks is constantly fading, in favour of the Transformers architectures. Vaswani et al. were, indeed, able to overcome the recurrency issues by applying a self-attention mechanism. The idea behind the attention mechanism was first introduced in the computer vision domain [17]. However, for attention models, we usually refer to structures like the neural machine translation introduced by Bahdanau et al. [18]. A

Transformer model, as proposed by Vaswani et al., consists of an encoder-decoder architecture. The main features of each structure are: to be highly parallelizable, thanks to the (multi-head) attention mechanisms and point-wise fully-connected layers; to be able to capture a long-term dependency, thanks to the attention mechanisms and the positional encoding. Such features allowed some research groups to exploit this kind of architecture to develop Language Models from large size unlabeled corpora. Examples of this kind of models are GPT [19] and its 1.5 and 17 billion parameters successors GPT-2/3 [20, 21] from $OpenAI$, $XLNET$ [22], $BERT$ [23], in its versions $Base$ (110 millions parameters) and $Large$ (340 millions parameters), and its optimized variants RoBERTa [24] and DistilBERT [25] (the latter, counting "only" 66 millions parameters). Most of the Transformer-based models are available through the *transformers* package from $HuggingFace$ [26]. In addition, the *transformers* library provides the pre-trained versions of the models themselves. This is particularly useful from a Transfer Learning paradigm [27] point of view. Those LMs were pre-trained on a very large amount of unlabeled text in a task-agnostic manner, and can therefore be fine-tuned for a specific task without training them from scratch. This kind of pipeline has already been shown to be very powerful: models have been effectively fine-tuned to a large variety of NLP tasks, both token-, sentence- and document-level tasks (such as the GLUE benchmark [28]), reaching the state-of-the-art performance in just a few epochs of training. In many cases, they overcome the performance of fine-tuned RNN-based LMs such as $ELMO$ [29] and $ULMFiT$ [30].

2.3. Hierarchy in Transformer Models

One of the greatest limitations of the Transformer-based models is to be limited to input of a fixed length of text, usually less than a few hundred tokens, even if they have the potentiality to learn longer-range context dependencies. This is due to the computational and memory requirements of the self-attention mechanism, which quadratically grows with the number of tokens in the sequence. The simplest approach to use for long document classification tasks with Transformers is, therefore, the truncation of the document. This obviously may lead to a significant loss of information. Trying to overcome this issue, some groups of researchers developed an extension of models like BERT. Such extensions usually exploit a hierarchical architecture, in which a classifier is built on a first Transformer, basing its prediction upon the representations of some chunks of text given in input to the Transformer model. For example, in [31] two kinds of architecture were investigated: RoBERT and ToBERT. They build each model upon stacked representation in output to a first BERT layer. In RoBERT, a Recurrency over BERT was implemented using an LSTM layer and two fully-connected layers. In ToBERT, another Transformer was used over BERT, substituting the LSTM layer with a 2-layers Transformer. At a cost of a greater computational cost, ToBERT showed better performance on some evaluated tasks, especially on the one dataset consisting of longer documents. For both models, each document was divided into chunks counting 200 tokens, with an overlap of 50 tokens for consecutive chunks. Inspired by this work, in [32] documents were divided into chunks of 512 tokens (with 50 overlapping tokens within consecutive segments), and an investigation on the merge method was conducted. In particular, the classification was based on the most representative vector (the one with the highest norm), on the average of all the vectors, and on a representation built through a 1D Convolutional layer. The main advantage of the two

merging methods is their efficiency, even more than the one method proposed in the previous paper. Closer to our task, there is the work in [33], where HIBERT, a hierarchical transformer (again, based on BERT) was first pre-trained in an unsupervised fashion and then fine-tuned on a supervised extractive summarization task, where all the sentences of each document are labelled as belonging or not to the summary of that document. Following this work, in [34] proposed to pre-train a hierarchical transformer model with a masked sentence prediction (in which the model is required to predict a masked sentence) and a sentence shuffling tasks (in which the model is required to predict the original order of the shuffled sentences). Then, also using the self-attention weights matrix (obtained by averaging over the heads for each layer and then averaging over the layers), the hierarchical pre-trained encoder is used to compute a ranking score for the sentences. The top-3 sentences are then used to constitute the summary. To the best of our knowledge, this last work is the closest to our, exploiting the attention weights of a hierarchical transformer model to generate a ranking useful to extractive summarization. However, this last model was used with the aim to generate summaries in an unsupervised manner, while we aimed to collaterally generate summaries that explain the decision of a hierarchical model in a task of document classification.

2.4. Attention as Explanation

In the recent literature, various works proposed to analyse the attention patterns of the Transformer architecture to have an insight on how such a model works. In [35] the author proposed a useful visualization tool, named $BertViz$. This tool provides an interactive interface to visualize attention weights between tokens for every attention head in every layer. Through this tool the author was able to find that some particular heads (in some particular layer) may capture lexical features such as verbs, acronyms, and heads related to coreference resolution, also showing the eventuality for such heads to also encode gender bias. Another kind of visualization tool for the attention weights is the attention (heat-)map. Using these maps, the authors in [36] found patterns that are consistent with the previous ones. In details, they divided the patterns in 5 categories: vertical (which mainly corresponds to attention to the delimiter tokens), diagonal (attention to previous/next word), a mix of these two, block (intra-sentence attention), and heterogeneous (said, no distinct structure). In this work, also an heads/layers disabling study was conducted, showing that in some cases a pruning strategy does not lead to a drop in the performance. Sometimes, this even leads to an increase in performance. Besides these two, other studies have been conducted showing that the self-attention heads allow BERT, as other Transformer models, to capture linguistic features, such as anaphora [37], subject-verb pairings [38] (then extended by [39]), dependency parse trees in encoder-decore machine translation models [40, 41], part of speech tags [42], dependency realations and rare words [43]. However, in our study, we did not aim to reach an explanation of how the Transformer model deals with such features, but we aimed to reach an interpretation of the document classification given by the model. Talking about this paradigm, there are various works that focus on the weights of the attentional layer in Transformers [44] or other kind of network, such recurrent or convolutional, to highlight the words or n-grams in the text that are the most relevant for the decision. Regarding the sentiment analysis task, authors in [45] observed a strong interaction between neighboring words visualizing the attention matrix of a Transformer-like network.

Furthermore, in [46], the authors of the work discussed the use of attention scores from an attention layer as a good and less computationally burdensome alternative to external explainer models like *LIME* [47, 48] and *Integrated Gradients* [49] methods. However, the result of such method is, again, to highlight parts of the discourse to get an insight on which words the model focus the most. This kind of approach does not lead to an actual interpretative summary, that may be more easily readable and therefore interpretable.

3. Materials and methods

To benchmark our models, we used the *IMDB Large Movie Review Dataset*. Such dataset consists of $50K$ movie reviews written in English and collected by [7]. Those reviews (no more than 30 reviews per movie) where highly polarized, as a negative review corresponds to a $score \leq 4$ (out of 10), and a positive one has a $score \geq 7$. We downloaded the data through the *Tensorflow*[1] *API*. The data is already divided in two equivalent sets, one for training and one for testing (plus $50K$ unlabelled reviews that one might used for unsupervised learning, not used in this work). Each of the subset presents a $50 : 50$ proportion between negative and positive examples. To assess the explainability of our methods we randomly extracted a total of 150 reviews, divided in two subsets, 50 from the training set and 100 from the test set. Documents were chosen maintaining the proportion between the two classes, ensuring that both the models can correctly classify them. Four annotators were instructed to select the three most important sentences in each document. To make such a choice, the annotator is allowed to look at the sentiment of the document. Obviously, to do a fair comparison, each document is restricted to a maximum of 15 sentences. To evaluate the agreement for the annotators, we calculated the so call *Krippendorff's alpha*. First proposed by Klaus Krippendorff [50], to which it owes its name, it is a statistic measure of the inter-annotator agreement/reliability. The strength of this index is to be applicable to any number of annotators, no matter to missing data, and can be use on various levels of measurement, such as binary, nominal and ordinal. This measure may be calculated as follows[2]: $\alpha = 1 - \frac{D_o}{D_e}$, where D_o is the disagreement $observed$, and D_e is the disagreement $expected$ by chance. Since the Krippendorff's alpha is calculated by comparing the pairs within each unit, those samples presenting at most one annotation are eliminated. However, in this case each sample (sentence) is automatically annotated as within the three most important sentences or not. Hence, such elimination phase was not required. Values of α less then 0.667 are often discarded, while values above 0.8 are often considered as ideal [51, 52]. Anyway, except for $\alpha = 1$, we could say that there is no such thing as a magical number as a threshold for this kind of analysis, especially for tasks as much subjective as this one. In our case, $\alpha_{training} = 0.47$ and $\alpha_{test} = 0.61$.

3.1. Models

Here we illustrate the two proposed architectures. In order to provide a visual explanation of them, we report the simplified schemes in Fig. 1.

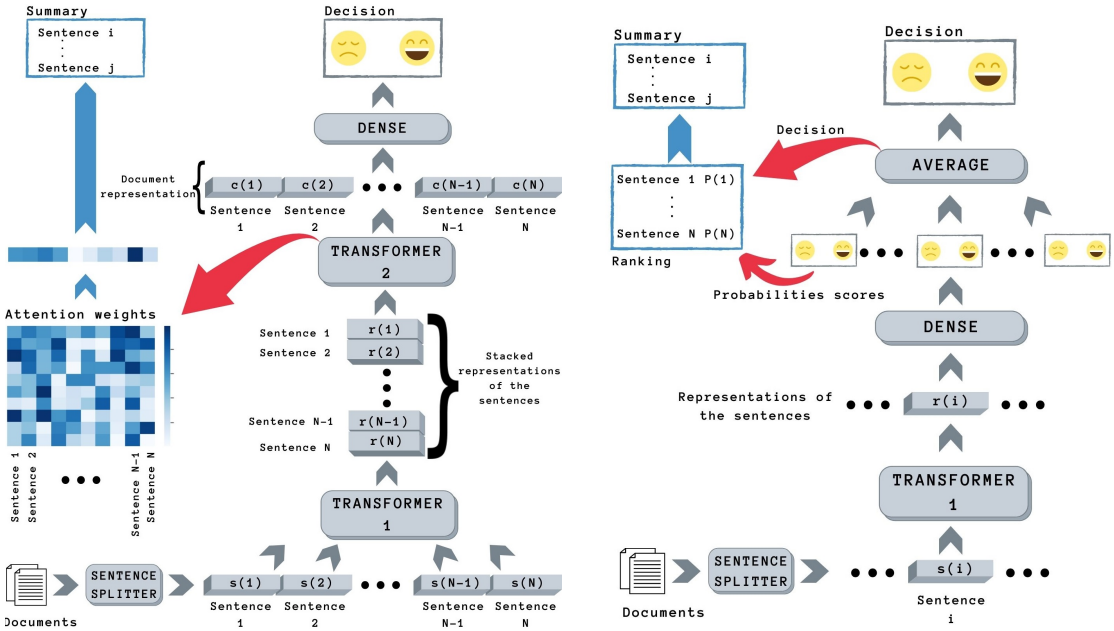
[1] www.tensorflow.org/datasets/catalog/imdb_reviews
[2] https://github.com/foolswood/krippendorffs_alpha

Explainable Hierarchical Transformer (ExHiT) The first model exploits a hierchical architecture, consisting of two Transformers ($T1$ and $T2$) in cascade (Fig. 1a). Because of its nature, we like to refer at this as *ExHiT*, the ***Ex**plainable **Hi**erarchical **T**ransformer*. The input of the first Transformer is a sequence of t tokens, while the output is an embedding representation of that sequence. Each sequence represents one of the N sentences $\{s_1...s_N\}$ in which the document is divided. If a document can be divided in just $m \leq N$ sentences, then $N - m$ empty sentences (just the special tokens) are added to the document. After $T1$ has elaborated the N sequences, the new generated representations $\{r_1...r_N\}$ are stacked together to become the input of $T2$. $T2$ then outputs a contextual representation c_i for the i-th sentence that depends on the other sentences ($c_i = f(r_1...r_N)$). By merging these contextual representations we obtain an unique document representation $d = U(c_1...c_N)$. In this work, we investigated three merging strategies: by concatenation: $U(.) = Concat(.)$; by averaging: $U(.) = Avg(.)$; by masked averaging: $U(c_1...c_N) = Avg(c_1...c_m)$ with $m \leq N$, for which $\{s_{m+1}...s_N\}$ is the set of added empty sentences; by the application of a Bidirectional LSTM: $U(.) = BiLSTM(.)$. Then vector d is given as input to a classification layer. In this work, such layer consists of a fully-connected dense layer for the binary classification task. Other than the contextual representations, we were able to retrieve from $T2$ also the self-attention weights for each head of each layer inside the transformer itself. To give more importance on the interpretability of the model instead of the performance, $T2$ consists only of two layers and just one head per layer. In this way, it is easier to extract valuable information. By averaging the attention weights associated with a specific sentence, we extracted the score of that sentence, that tells us how much it is informative for the task of classification. The sentences are ranked through such a score, and the most important ones are then selected to provide an extractive summary of the document. Such summary serves than as the explanation of the model decision.

Sentence Classifier Combination model (SCC) This second model has a simpler architecture, requiring just one Transformer model in its pipeline. The input of this Transformer is again a sequence of t tokens, i.e. the single sentence s_i. And again, its output is still a new representation r_i of that sentence. Such representation is given in input to a *Dense* layer to classify the sentiment of the sentence, outputting two probability scores, one for each class. Then the negative scores are averaged together, and the same for the positive ones, to get a final rating for each class. The prediction of the overall document sentiment will be given by whoever has the greatest final score. Knowing the decision of the model, the sentences are ranked by the inherent probability score. Then, the most relevant ones are extracted to build the summary of document, serving as an explanation of the model decision.

3.2. Experiments

Following we listed the main features of the two models used in the experiment's session: **T1**: for a fair comparison, the first transformer model was the same for both the architectures; we opted to use the pre-trained version of RoBERTa [24]; **T2**: we used a transformer with two layers, one head per layer; this choice was motivated to facilitate the explainability phase; **N**: the maximum number of sentences per document was set to 15; by this way, we ensured that the 75% of the training documents were elaborated in their entirety; **t**: the maximum number of



(a) Hierarchical Transformers model

(b) Sentence Classifier Combination model

Figure 1: A visual schematic representation of the two proposed explainable models.

tokens per sentence was set to 32, comprehensive of the two special delimiter tokens; by this way, we ensured that the 75% of the training sentences were elaborated in their entirety. Besides the two models, we implemented a pre-processing phase consisting of the replacement of the tokens '<\br><\br>' with the newline character, and, obviously, a sentence splitting step. We used the sentence tokenizer provided by NLTK. Furthermore, for documents that do not reach N number of sentences, empty sentences (consisting of just the special tokens) were added up to N. Similar reasoning was applied to sentences that do not reach t number of tokens: in these cases, the sequences were zero-padded on the right. The first model was jointly trained with a batch size of eight documents for the task of document classification. The second model was trained on the task of single sentence classification, with a batch size of 240 sequences.

4. Results

The proposed models were evaluated for both sentiment analysis and explainability outcomes. In Tab. 1 we reported the best sentiment analysis results achieved in terms of accuracy, precision and recall per class. For the *ExHiT* model, various proposed merging strategies were tested. As the accuracy column highlights, changing the merging strategy does not significantly affect classification performance. Furthermore, the *SCC* model seems to achieve slightly better overall performance. However, it is interesting to notice that *SCC* result particularly good for the precision for the negative class and the recall for the positive one, while achieving the worst performances for their counterpart metrics, for which the best results are obtained by

Model	Merging strategy	Accuracy (%)	Precision (%)		Recall (%)	
			Neg	Pos	Neg	Pos
ExHiT	Concatenation	92.59	90.97	**94.34**	**94.56**	90.62
	Average	92.35	92.18	92.51	92.54	92.15
	Masked Average	92.77	92.07	93.49	93.60	91.94
	BiLSTM	92.34	90.97	93.80	94.01	90.67
SCC	-	**93.51**	**95.42**	91.75	91.40	**95.62**

Table 1
Sentiment analysis results in terms of accuracy, and precision and recall per class.

Model	Merging strategy	Agreement at least 1 Precision (%)		Agreement at least 2 Precision (%)		Agreement at least 3 Precision (%)	
		test	train	test	train	test	train
ExHiT	Concatenation	53.82%	55.88%[a]	49.15%	45.00%	46.63%	46.45%
	Average	58.04%	57.82%	50.42%	45.92%[1]	45.29%	41.84%
	Masked Average	53.15%[a]	55.79%	45.97%[a]	44.92%	40.66%	39.80%
	BiLSTM	55.51%[a]	55.85%	49.05%[a]	45.24%[a]	43.38%[a]	39.95%
SCC	-	70.74%	65.61%	65.22%	57.83%	55.22%	47.52%

Table 2
Explainability performance in terms of precision (averaged over all documents) for different annotators agreements, evaluated on both the annotated documents from training and test sets.

ExHiT using the concatenation merging strategy. Following the same structure, in Tab. 2 we reported the explainability outcomes in terms of precision averaged over all the documents. The performances are reported for different annotators agreements, i.e. we built summaries by grouping the sentences for which at least one, two or three out of the four annotators judged them among the most important ones. This implies that some annotators summaries may contain more than three sentences ($N > 3$, especially in the first case) or less than three sentences ($N < 3$, especially in the latter case). So, we extracted the first N sentences in the machines ranking and evaluated the overlap of these summaries with the annotators' ones. About the *ExHiT* performance, the results of the best layer are reported. In general, the ranking from the first layer slightly outperformed the rankings from the last layer[1] and the rankings obtained by averaging both layers[a]. Furthermore, the empty sentences were removed by the machine rankings. Even with these adjustments, the *ExHiT* explainability results are lower than those of *SCC*, with respect to all the merging strategies. This outcome may be the result of an influence of the task on the two models: it may be noticed that the task the second model accomplishes is closer to the one performed by the annotators. This may therefore result in helping the model in the explainability task. Another interesting fact that may be noticed is that the average merging strategy leads to better performance than the masked one, especially with respect to the test set ($\sim +5\%$). This seems to suggest that masking the empty sentences from the average combination does not help the model to better understand the task.

5. Discussion and Conclusion

In this work we proposed a new approach to achieve an explainable document classification, also proposing two different kinds of models. We evaluated them on a sentiment analysis task, also providing an annotated dataset to test the explainability performance. However, both

underlying architectures allow their easy adaptation in any document classification task (e.g. topic classification). Both models have achieved good classification results, not so far from the state-of-the-art on the *IMDB* dataset, while also performing an explanation in the form of a summary. To the best of our knowledge, this is the first attempt to build a document classification paradigm of models that generate an extractive summary in order to provide an easy to interpret explanation to the user. Such models may be implemented in some application systems, for example customer care or market research tools. Indeed, while sentiment analysis may help to mark customer messages and reviews, the explainability part may be helpful to get quick insights about strengths and weaknesses of some product or service.

Further research works may evaluate such models in different classification tasks. Sentiment analysis is a task that particularly relies on the lexical meaning of individual sentences. Testing a different kind of task may show *ExHiT* outperforming the second model because able to get more insights from the context of the document. Furthermore, it would be interesting to exploit the potential of both models to be able to operate on tasks involving longer documents, that is a sort of limitation for traditional Transformer architectures.

References

[1] D. Gunning, Explainable artificial intelligence (xai), Defense Advanced Research Projects Agency (DARPA), nd Web 2 (2017).

[2] D. Gunning, D. Aha, Darpa's explainable artificial intelligence (xai) program, AI Magazine 40 (2019) 44–58.

[3] A. B. Arrieta, et al., Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai, Information Fusion 58 (2020) 82–115.

[4] O. Loyola-Gonzalez, Black-box vs. white-box: Understanding their advantages and weaknesses from a practical point of view, IEEE Access 7 (2019) 154096–154113.

[5] M. Danilevsky, et al., A survey of the state of explainable ai for natural language processing, arXiv preprint arXiv:2010.00711 (2020).

[6] A. Vaswani, et al., Attention is all you need, in: Advances in neural information processing systems, 2017, pp. 5998–6008.

[7] A. Maas, et al., Learning word vectors for sentiment analysis, in: Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies, 2011, pp. 142–150.

[8] W. S. El-Kassas, et al., Automatic text summarization: A comprehensive survey, Expert Systems with Applications (2020) 113679.

[9] M. I. Jordan, Serial order: A parallel distributed processing approach, in: Advances in psychology, volume 121, Elsevier, 1986, pp. 471–495.

[10] J. L. Elman, Finding structure in time, Cognitive science 14 (1990) 179–211.

[11] M. Schuster, K. K. Paliwal, Bidirectional recurrent neural networks, IEEE transactions on Signal Processing 45 (1997) 2673–2681.

[12] Y. Bengio, et al., Learning long-term dependencies with gradient descent is difficult, IEEE transactions on neural networks 5 (1994) 157–166.

[13] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (1997) 1735–1780.
[14] K. Cho, et al., Learning phrase representations using rnn encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078 (2014).
[15] M. Sundermeyer, et al., Lstm neural networks for language modeling, in: Thirteenth annual conference of the international speech communication association, 2012.
[16] M. Sundermeyer, et al., From feedforward to recurrent lstm neural networks for language modeling, IEEE/ACM Transactions on Audio, Speech, and Language Processing 23 (2015) 517–529.
[17] H. Larochelle, G. E. Hinton, Learning to combine foveal glimpses with a third-order boltzmann machine, in: Advances in neural information processing systems, 2010, pp. 1243–1251.
[18] D. Bahdanau, et al., Neural machine translation by jointly learning to align and translate, arXiv preprint arXiv:1409.0473 (2014).
[19] A. Radford, et al., Improving language understanding by generative pre-training, 2018.
[20] A. Radford, et al., Language models are unsupervised multitask learners (2019).
[21] T. B. Brown, et al., Language models are few-shot learners, arXiv preprint arXiv:2005.14165 (2020).
[22] Z. Yang, et al., Xlnet: Generalized autoregressive pretraining for language understanding, in: Advances in neural information processing systems, 2019, pp. 5753–5763.
[23] J. Devlin, et al., Bert: Pre-training of deep bidirectional transformers for language understanding, arXiv preprint arXiv:1810.04805 (2018).
[24] Y. Liu, et al., Roberta: A robustly optimized bert pretraining approach, arXiv preprint arXiv:1907.11692 (2019).
[25] V. Sanh, et al., Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter, arXiv preprint arXiv:1910.01108 (2019).
[26] T. Wolf, et al., Transformers: State-of-the-art natural language processing, in: Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, 2020, pp. 38–45.
[27] S. J. Pan, Q. Yang, A survey on transfer learning, IEEE Transactions on knowledge and data engineering 22 (2009) 1345–1359.
[28] A. Wang, et al., Glue: A multi-task benchmark and analysis platform for natural language understanding, arXiv preprint arXiv:1804.07461 (2018).
[29] M. E. Peters, et al., Deep contextualized word representations, arXiv preprint arXiv:1802.05365 (2018).
[30] J. Howard, S. Ruder, Universal language model fine-tuning for text classification, arXiv preprint arXiv:1801.06146 (2018).
[31] R. Pappagari, et al., Hierarchical transformers for long document classification, in: 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), IEEE, 2019, pp. 838–844.
[32] A. Pelicon, et al., Zero-shot learning for cross-lingual news sentiment classification, Applied Sciences 10 (2020) 5993.
[33] X. Zhang, et al., Hibert: Document level pre-training of hierarchical bidirectional transformers for document summarization, arXiv preprint arXiv:1905.06566 (2019).

[34] S. Xu, et al., Unsupervised extractive summarization by pre-training hierarchical transformers, arXiv preprint arXiv:2010.08242 (2020).
[35] J. Vig, A multiscale visualization of attention in the transformer model, arXiv preprint arXiv:1906.05714 (2019).
[36] O. Kovaleva, et al., Revealing the dark secrets of bert, arXiv preprint arXiv:1908.08593 (2019).
[37] E. Voita, et al., Context-aware neural machine translation learns anaphora resolution, arXiv preprint arXiv:1805.10163 (2018).
[38] Y. Goldberg, Assessing bert's syntactic abilities, arXiv preprint arXiv:1901.05287 (2019).
[39] T. Wolf, Some additional experiments extending the tech report" Assessing BERTs syntactic abilities" by Yoav Goldberg, Technical Report, Technical report, 2019.
[40] J. Hewitt, C. D. Manning, A structural probe for finding syntax in word representations, in: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 2019, pp. 4129–4138.
[41] A. Raganato, et al., An analysis of encoder representations in transformer-based machine translation, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, The Association for Computational Linguistics, 2018.
[42] J. Vig, Y. Belinkov, Analyzing the structure of attention in a transformer language model, arXiv preprint arXiv:1906.04284 (2019).
[43] E. Voita, et al., Analyzing multi-head self-attention: Specialized heads do the heavy lifting, the rest can be pruned, arXiv preprint arXiv:1905.09418 (2019).
[44] L. Franz, et al., A deep learning pipeline for patient diagnosis prediction using electronic health records, arXiv preprint arXiv:2006.16926 (2020).
[45] G. Letarte, et al., Importance of self-attention for sentiment analysis, in: Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP, 2018, pp. 267–275.
[46] F. Bodria, et al., Explainability methods for natural language processing: Applications to sentiment analysis (discussion paper) (2020).
[47] M. T. Ribeiro, et al., " why should i trust you?" explaining the predictions of any classifier, in: Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining, 2016, pp. 1135–1144.
[48] Y. Zhang, et al., " why should you trust my explanation?" understanding uncertainty in lime explanations, arXiv preprint arXiv:1904.12991 (2019).
[49] M. Sundararajan, et al., Axiomatic attribution for deep networks, in: International Conference on Machine Learning, PMLR, 2017, pp. 3319–3328.
[50] K. Krippendorff, Estimating the reliability, systematic error and random error of interval data, Educational and Psychological Measurement 30 (1970) 61–70.
[51] K. Krippendorff, Content analysis: An introduction to its methodology (2 nd thousand oaks, 2004.
[52] K. Krippendorff, Reliability in content analysis: Some common misconceptions and recommendations, Human communication research 30 (2004) 411–433.