Text Classification and post-hoc XAI approach for Text Summarization on PubMed

Ermellino Andrea, Kolyszko Matteo

Department of Informatics, Systems and Communication, DISCo , Università degli Studi di Milano Bicocca
Piazza dell'Ateneo Nuovo, 1, 20126 Milano MI

Abstract: This paper presents a pipeline for text classification and summarization using multi-class classification to identify sentences in one of five categories. The classification study is based on reference paper [1], which proposed four different models and two different representations for classifying sentences. The task is modeled as a multi-class classification problem and two datasets are obtained through preprocessing. Two text representations, BoW (tf-idf) and Sentence Embeddings (Universal sentence encoder), are used for classification. Results from seven different models are compared to those of the reference paper. Additionally, the paper proposes an alternative Explainable AI approach compared to the one proposed by [2] for text summarization, using the post-hoc approach and the XAI model SHAP (SHapley Additive exPlanations). The results of this approach are also discussed in the paper.

Key words: Text Classification, Text Summarization, XAI

Contents

1 Introduction

In this document we propose the results obtained in Text Classification and Text Summarization tasks performed on PumMed dataset. For both tasks we considered three papers (two without code), [1] for classification task, [2] and [3] for summarization task, on which we based the study carried out and finally compared the results.
PubMed is a database maintained by the National Library of Medicine (NLM) that contains abstracts and citations for biomedical literature, including journal articles, books, and conference proceedings. For our tasks, we used portions of PubMed database according to the ones used by [1], [2] and [3].

2 Text Classification

This section of the document describes the entire pipeline of the Text classification task, starting from the idea, to processing, to final results obtained.

2.1 Reference Paper

The analysis we propose is based on [1] work. They proposed four different models and two different representation in order to classify sentences.

2.2 Pipeline

The task was modeled as multi class classification problem, figuring out whether a sentence belongs to one of the five categories. After preprocessing two dataset were obtained, one with the stopwords and one without. Two different text representation were chosen in order classify the sentences, BoW (tf-idf) and Sentence Embeddings (Universal sentence encoder). In total seven different model were develop for this task.

2.3 Preprocessing

The process starts by segmenting raw documents into text sentences each with its own label associated. Then, the process follows by removing punctuation.

2.4 TF-IDF

As mentioned earlier, two different types of representations were used. In the first case, TF-IDF was chosen because we wanted to use the same method as the paper previously mentioned.
The first baseline is a classifier based on Naive Bayes (MultinomialNB) using bi-grams features extracted from the current sentence: it does not use any information from the surrounding sentences.
The second baseline is a classifier based on Support Vector Machine (LinearSVC) using tri-grams features extracted from the current sentence: it does not use any information from the surrounding sentences.

Classifier	Precision	Recall	F-Measure
Dernoncourt et al.[1]	85.8	86.1	85.9
MultinomialNB	81.1	81.2	81.1
LinearSVC	84.3	84.1	84.2

Table 1: TF-IDF classifier comparison

2.5 Universal Sentence Encoder

The second type of representation used is the Universal Sentence Encoder which is used for encoding sentences into embedding vectors.
The first baseline is a Forward ANN: it computes sentence embeddings for each sentence, then classifies the current sentence given a few preceding sentence embedding.
The second baseline is a bi-ANN it computes sentence embeddings for each sentence, then classifies the current sentence given a few preceding sentence embedding.

Classifier	Precision	Recall	F-Measure
Dernoncourt et al.[1]	/	/	88.4
Forward ANN	81.8	82.1	82.0
Dernoncourt et al.[1]	91.7	91.6	91.6
bi-ANN	82.1	82.2	82.2

Table 2: ANN classifier comparison

3 Text Summarization

This section of the document describes the entire pipeline of the Text Summarization task, starting from the idea, to processing, to final results obtained.

3.1 Reference Paper

The analysis we propose is based on [2] work. They proposed a Text Summarization approach using Explainable AI algorithms in order to obtain outputs that can be interpreted and justified. To be more precise, they used GAMI (Generative Additive Model with Interaction) models, GAMI-Net and Explainable Boosting Machine, that approach the task in *ante-hoc* way. The *ante-hoc* approach consists in building a white-box model capable of produce an output that is interpretable by design, so the explainability of the model is built-in.
Our proposal is to use a different XAI approach to address the task, the *post-hoc* approach: it consists in the usage of a XAI model that, given a black-box base model that produce outputs for the summarization task, tries to explain how the base model worked to produce the outputs. This XAI model trains a surrogate white-box model that ”learns” and ”explain” the outputs of black-box base model, optimizing a fidelity function (maximize in case of accuracy metrics as fidelity function, minimizing in case of loss metrics as fidelity function). So the base model explainability is not provided by design but it’s constructed via usage of a surrogated white-box model.
The XAI model used to address summarization task is SHAP (SHapley Additive exPlanations), a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

3.2 Pipeline

The task was modeled as a binary classification problem, figuring out whether a sentence is important (and therefore to be part of the summary) or not.

In order to compare the results obtained with the ones proposed by [2], preprocessing steps are the same ones proposed by them.
After preprocessing data, six different features are extracted from sentences to become inputs vectors that are used to train and predict. The black-box classifier used to predict is a Random Forest, then using SHAP explainer it was made possible to interpret the decisions made by the base model.
In order to perform the classification, labels generated and made public by [3] were considered.
Due to very high computational times, only a portion of the original dataset was used (11%, 12k documents, 2M sentences).

3.3 Preprocessing

The preprocessing step is responsible for turning raw documents into sequences of sentences, capturing useful information for future feature extraction. The process starts by segmenting raw documents into text sentences, which are split in tokens. Then, the process follows by removing punctuation and stopwords, word tagging and stemming. After preprocessing, sentences correspond to lists of their respective words that are forwarded to the feature extraction step. We perform most of the NLP preprocessing with the Python library spaCy, with the exception of stemming step which is done using the NLTK SnowballStemmer.

3.4 Feature Extraction

After sentences are preprocessed, six different features are extracted from sentences to become inputs vectors $x = \{x_1, x_2, ..., x_6\}$ that are used to train and predict. The feature computations are formulated as described below.

3.4.1 TF-ISF

TF-ISF is a variant of the TF-IDF method applied at sentence level for text summarization. The idea is to compute a score to each sentence based on term importance and descriptiveness inside the document, which are measured by term frequency (TF) and inverse sentence frequency (ISF) of the terms. We use bigrams TF-ISF, so each sentence s_i of a document receives a salience score (Equation 1) based on its term bigrams *b*:

$$w(s_i) = \sum_{j=1}^{J_i} [F(b_j) \times \log(\frac{n}{n_{b_j}})], \tag{1}$$

$$x_1(s_i) = \frac{w(s_i)}{max(w(s_i))} \tag{2}$$

where $F(b)$ is the frequency of *b* in the document, *n* is the number of sentences in the document, n_b is the number of sentences of the document in which *b* occurs and J_i is the number of bigrams in s_i.

3.4.2 Position

This feature indicates how early or how late sentences appear may give important information about their relevancy. The position feature (Equation 3) represents the sentence position inside the document, where p_i is the position of sentence s_i:

$$x_2(s_i) = \frac{p_i}{n} \tag{3}$$

3.4.3 Length

The length feature (Equation 4) is calculated based on the length of sentence s_i in terms of the maximum sentence length. The length feature allows the model to learn the relationship between sentence length and relevancy:

$$x_3(s_i) = \frac{number\ of\ terms\ in\ sentence\ s_i}{\max\left(number\ of\ terms\ in\ a\ sentence\ \right)} \tag{4}$$

3.4.4 Proper Nouns and Numerical

The individual ratio of proper noun and numerical terms in the sentence s_i may indicate the presence of relevant information. We calculate these features as follows:

$$x_4(s_i) = \frac{number\ of\ proper\ nouns\ in\ s_i}{number\ of\ terms\ in\ s_i} \tag{5}$$

$$x_5(s_i) = \frac{number\ of\ numerical\ terms\ in\ s_i}{number\ of\ terms\ in\ s_i} \tag{6}$$

3.4.5 Sentence-Sentence similarity

The sentence-sentence similarity denotes how close a sentence is to other sentences in the document. We calculate this feature using cosine similarity *c* as in Equation 7:

$$x_6(s_i) = \frac{\sum_{j=1}^{n} c(s_i, s_j)}{\max(\sum_{j=1}^{n} c(s_k, s_j))}, i \neq j \tag{7}$$

The similarity *c* is computed using TF-IDF weighting scheme.

3.5 Sentence scoring and selection

Once having calculated the feature vector for every sentence of a given input document, we use a Random Forest to predict the probability of a sentence to be important or unimportant. Obtained the probabilities, the sentences of the document are ranked according to the probability of being important. Then, the first *n* sentences are selected to be part of the summary, respecting the token limit set at 200.

3.6 SHAP Explanation

SHAP uses a concept called Shapley values from cooperative game theory to assign importance scores to each feature of a dataset. It can explain the prediction obtained for a single instance, showing the contribution of each feature to the prediction, but can also give a global explanation of how the model has predicted several instances.
In this particular case, plotting the contribution of features for a given prediction we can see how and with what intensity a feature has influenced a prediction:

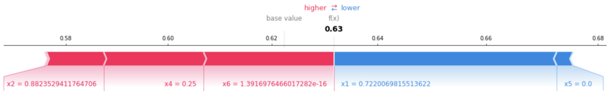

Figure 1: Force Plot of single instance prediction explanation

It is possible to see in red the features that pushed the prediction to the positive class (important sentence), in blue there are the features that pushed the prediction to the negative class (not important sentence). In Figure 1 example, the instance was predicted as positive, so an important sentence. Value of features x_2, x_4, x_6 pushed the prediction to the positive class, instead value of features x_1, x_5 pushed the prediction to the negative class.
SHAP gives also an overview of the overall importance of each feature in a dataset, by plotting the summary it is possible to see:

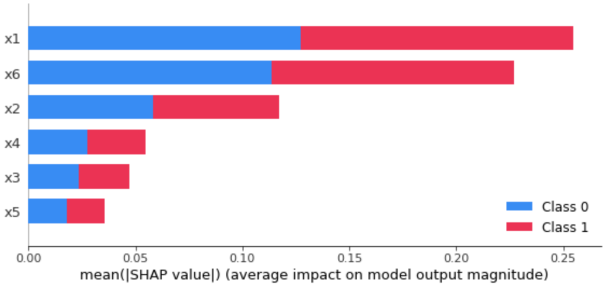

Figure 2: Summary Plot in bar form of multiple instance prediction

Figure 2 shows the importance of features in a sample of 10 predictions. The higher the bar, the more important the feature was in contributing to predictions, the color shows in which way its value pushed the prediction to positive/negative class.
SHAP gives a high level of explainability to the base black-box model, regardless of his way of computing predictions. In this case, the Random Forest with 100 estimators used to compute predictions was extremely not interpretable if taken as it is, with SHAP explainer we obtained instead a high level of interpretability.

3.7 Model Performance Comparison

In order to solve the unbalanced class problem, different resampling techniques have been tried: random undersampling, random oversampling and SMOTE. The random undersampling produced the best results, avoiding overfitting, but different combination of techniques can lead to even better results (SMOTE + undersampling, more data of minority class with SMOTE but avoids overfitting with undersampling). Random Forest performed as follows:

Model	Precision	Recall	F1
Random Forest + SHAP	70	70	70
EBM [2]	/	/	30.70
GAMI-Net [2]	/	/	30.76

This results shows that XAI *post-hoc* approach using a black-box base model can lead to good results, in this case better than using a XAI *ante-hoc* approach like GAMI-Net or EBM. Better results can be obtained using a much more powerful black-box model like a Deep Neural Network, we trained one with the following architecture:

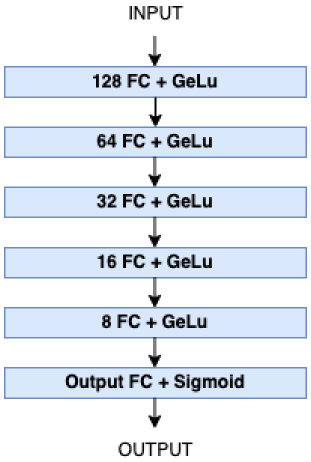

Figure 3: Deep Neural Network Architecture

This NN performs as follows:

Model	Precision	Recall	F1
Deep Neural Network	71	71	71

Due to extremely high computational costs of SHAP + NN, we didn't managed to get explanations for predictions made by this classifier, so we used Random Forest to perform the summary creation step.
The summary creation step consists in processing a given document in input, computing features vector for its sentences, predicting the probability of being important or not important for each sentence, ranking the sentences by the probability of being important and selecting the first sentences with respect to the token limit of summarty set to 200.
Taken as input 90 "hidden" documents (90 documents that the model has never seen before), we created the summaries following the steps described above and evaluated them using ROUGE-n F-Measure metric:

Model	ROUGE-1 (%)	ROUGE-2 (%)	ROUGE-l (%)
Random Forest + SHAP	34.97	13.86	31.53
GAMI-Net [2]	39.78	13.92	34.57
EBM [2]	38.86	13.96	34.65

Obviously the metrics described above are calculated on a smaller set of summaries than those used by [2], but the comparison shows how the 3 different models performs quite equally.

4 Conclusions and Future Developments

Results showed how the XAI *post-hoc* approach can lead to good results, preserving the power of black-box model and giving a high level of explaianability. *Ante-hoc* approach force the usage of a predefined white-box model, even if they are still powerful models they have not the computational ability proper of the most powerful black-box models (such as DNN).
Obviously this analysis could be a starting point for future developments, try different combination of resampling techniques could lead to better performance of models, even try different Neural Network architectures. Having more computational power available, train the model on increasingly larger portions of the dataset could lead to better results.

5 References

References

[1] Franck Dernoncourt and Ji Young Lee. *PubMed 200k RCT: a Dataset for Sequential Sentence Classification in Medical Abstracts*. Computation and Language (cs.CL); Artificial Intelligence (cs.AI); Machine Learning (stat.ML). 2017.

[2] Vinícius Camargo da Silva, João Paulo Papa, and Kelton Augusto Pontara da Costa. *Extractive Text Summarization Using Generalized Additive Models with Interactions for Sentence Selection*. São Paulo State University - UNESP, Bauru, Brazil, Dec. 2022.

[3] Xiao W. and Carenini G.: *Extractive summarization of long documents by combining global and local context*. Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 3011–3021, 2019.