BERT-based Conformal Predictor for Sentiment Analysis

Lysimachos Maltoudoglou L.MALTOUDOGLOU@ALBOURNE.COM
Machine Learning Research Group,
Albourne Partners Ltd, London, UK
Computational Intelligence Research Lab.,
Frederick University, Nicosia, Cyprus

Andreas Paisios A.PAISIOS@ALBOURNE.COM
Machine Learning Research Group,
Albourne Partners Ltd, London, UK
Computational Intelligence Research Lab.,
Frederick University, Nicosia, Cyprus

Harris Papadopoulos H.PAPADOPOULOS@FREDERICK.AC.CY
Computational Intelligence Research Lab.,
Frederick University, Nicosia, Cyprus
Machine Learning Research Group,
Albourne Partners Ltd, London, UK

Editor: Alexander Gammerman, Vladimir Vovk, Zhiyuan Luo, Evgueni Smirnov and Giovanni Cherubin

Abstract

We deal with the Natural Language Processing (NLP) task of Sentiment Analysis (SA) on text, by applying Inductive Conformal Prediction (ICP) on a transformers based model. SA, which is the interpretation and classification of emotions, also referred to as emotional artificial intelligence, can be set up as a Text Classification (TC) problem. Transformers are deep neural network models based on the attention mechanism and make use of transfer learning by being pretrained on a large unlabeled corpus. Transformer based models have been the state of the art for dealing with various NLP tasks ever since they were proposed at the end of 2018. Our classifier consists of the BERT model for turning words into contextualized word embeddings with parameters fine-tuned on the used corpus and a fully connected output layer for performing the classification task. We examine the performance of the underlying BERT model and the proposed ICP on the Large Movie Review dataset consisting of 50000 movie reviews. The results show that the good performance of the underlying classifier is carried on to the ICP extension without any substantial accuracy loss while the provided prediction sets are tight enough to be useful in practise.

Keywords: Conformal Prediction, Inductive CP, Sentiment Analysis, NLP, Transformers, BERT, Transfer learning, Text Classification,

1. Introduction

Sentiment Analysis (SA) or opinion mining, aims to extract subjective information such as opinions, attitudes and emotions from human language. Services such as social media platforms and online forums can provide an abundance of data where consumer and political preferences are contained. Analysing this data can lead to information on how to increase

market share and company profits or on how to increase popularity of leaders and influencers. SA is a task of Natural Language Processing (NLP), but it can also be applied on images or videos of humans to extract similar kind of information from facial expressions, gesture and body language. This is referred to as Visual SA and belongs to the field of Computer Vision.

Although SA has been attracting the interest of researchers and practitioners alike due to its potential practical applications (Liu, 2010), it was only after 2004 that the research on SA began to flourish. The reasons are found in the development of computer systems in the 90's that allowed computer-based SA as well as in the increased availability of large opinion text corpuses from the Internet after the 00s (Mäntylä et al., 2018). NLP's advancements in the past decade pushed SA research significantly making it now one of the most active research areas in computer science.

Approaches to the SA problem can be categorised based on the techniques used and the level of classification. The two main techniques used are Machine Learning (ML) and the Lexicon-based (Medhat et al., 2014). ML techniques use a variety of learning algorithms to determine the sentiment by training on a known dataset. Lexicon-based techniques extract the sentiment using the semantic orientation of words. Depending on the level of SA performed, these can be categorised as: document; sentence; and aspect-level SA. Document level SA extracts the sentiment of a whole document while sentence-level operates on each sentence within the text. Aspect level SA is based on the idea that an opinion is made-up by a sentiment on a target (Liu, 2012), and therefore it extracts sentiments with regards to certain entities or aspects of entities within the text. To demonstrate how aspect level SA serves the problem, consider the following example of a segment of a phone purchase review: "Although the battery charges really fast it doesn't last long". Different sentiments for different aspects of the entity "battery" co-exist within this text.

Document and sentence-level SA suffer from the limitation of providing general sentiments without discovering what exactly people liked and disliked. Aspect level SA consists of a set of NLP sub-tasks to deal with this limitation making it a challenging problem. Document and sentence-level SA are considered a Text Classification (TC) process, as text is automatically assigned to one or more tags out of fixed set of possible tags such as: "happy", "angry", "sad". TC is one of the basic NLP tasks and it comes into 3 flavors: the binary setting, where each text belongs to one out of two categories; the multi-class setting, where each text belongs to exactly one out of many categories; and the multi-label setting, where each text is assigned to one or more categories out of many categories.

NLP is a set of computational techniques combining linguistics, computer science and artificial intelligence that aim to enable computers to analyze, represent and reproduce human language. In particular, the higher purpose of NLP is to allow computers to perform a number of human-language related tasks such as categorization, translation, information extraction and text generation. In the 2010s NLP tasks marked a significant increase in performance (Goth, 2016), fueled by scientific advancements in ML along with the computation power boost provided from hardware. Advancements in ML were mostly driven by Artificial Neural Networks (ANNs) which are inspired from the function of the human brain. Current state of the art models in NLP make use of Transformers which were recently proposed as an alternative type of ANN structure to deal with NLP tasks.

Despite the fact that SA and TC problems have been extensively dealt with in the bibliography, the issue of providing real likelihoods for predictions being correct still persists and when such information is provided it can still be misleading, as shown by Papadopoulos (2013) and Antonis et al. (2015). This limitation is addressed by Conformal Prediction (CP), which provides prediction-sets for a pre-specified confidence level that are guaranteed to be well calibrated under the assumption of data exchangeability (Shafer and Vovk, 2007). CP can be very useful when opinion mining predictions need to satisfy a pre-specified level of confidence due to the sensitivity of the final conclusions, e.g. when a manufacturer needs to decide if investing more in a specific feature of his product would be beneficial for his business. In addition, there are situations in SA where the same or a similar conclusion is reached under different predictions e.g. when extracting emotions "happy" and "very happy". In this case prediction regions can be beneficial as well. A drawback of CP is that it is a computationally heavy process. One of the ways to address this problem is with Inductive CP (ICP) which offers the same guarantees as CP without any further assumptions.

In this study we investigate the use of ICP on combined with a TC classifier (the underlying model) based on BERT: Bidirectional Encoder Representations from Transformers. We are dealing with the binary TC problem of predicting the polarity (positive or negative sentiment) in movie reviews (document level SA). We measure the ICP performance and compare it with that of the underlying classifier using the results from the forced-prediction mode of CP. BERT is one of the first transformer-based models proposed and highly influential for future models. Our contribution is that we supplement the underlying transformer-based model predictions with reliable confidence information by utilizing the CP framework. To the best of our knowledge ICP has not been used on a transformer-based model before. We also experiment with two types of classifiers to measure their impact on the use of CP.

The paper is structured as follows: Section 2 presents a review of the existing literature on NLP and TC. Section 3 discusses CP and ICP in more detail and Section 4 outlines our approach for the particular problem. In Section 5 we describe our experimental set-up and present results which are further discussed with our conclusions in Section 6.

2. Literature Review

As a TC problem, document-level SA is primarily influenced by the work on NLP and particularly by developments on neural word embeddings that provide high quality features for the underlying classifiers. Word embeddings are dense, high dimensional vector representations of text able to attribute semantic similarities and analogies. Word embeddings have been increasingly popular because of the continuous performance improvements that their use has demonstrated in a wide range of basic NLP tasks, (Cambria et al., 2017), TC included, and can be categorised as Static and Dynamic (or Contextualized) word embeddings, as shown in Figure 1.

Static word embeddings are produced from shallow models, usually pre-trained. The Word2Vec method (Mikolov et al., 2013a), (Mikolov et al., 2013b)) was the first efficient static word embeddings as it offered a way to deal with each word in a collective way and not individually as was previously the case. Pennington et al. (2014), introduced the Global

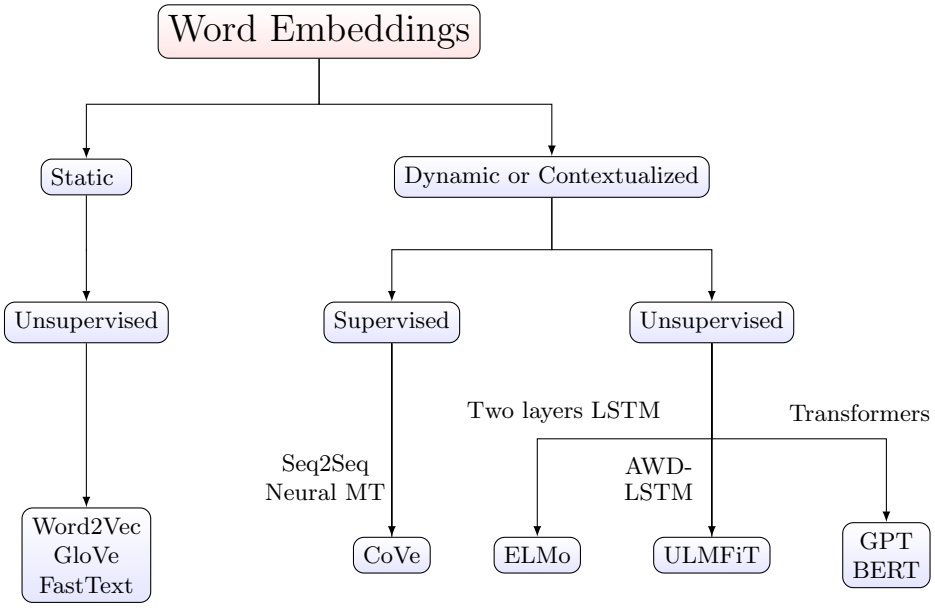

Figure 1: Categorization of recent Word Embedding models by type and learning algorithm

Vectors (GloVe) method which combined the architecture of Word2Vec with the statistical Latent Semantic Analysis (LSA) method, reducing their weaknesses. Joulin et al. (2016), proposed FastText, a method that uses subword information to deal with the issue of not having vector representations for words outside the vocabulary.

Developments that led to transformer-based models, made use of contextualized word embeddings. Transformers (Vaswani et al., 2017) are pre-trained language models based on sophisticated (deep) ANNs that make use of the *attention mechanism*, Sequence to Sequence (*Seq2Seq*) architecture and *transfer learning*. Language models aim to predict the next word in a sequence of words, given a number of words that precede it. The attention mechanism (Bahdanau et al., 2014) was proposed to incorporate into machines an approach similar to the way humans pay attention and recall memories. Seq2Seq models (Sutskever et al., 2014) were first proposed to deal with the Machine Translation (MT) problem and are composed of an encoder and a decoder. Under transfer learning view models are developed for a task where data is abundant and then reused as the starting point for models in specific tasks.

The first contextualised word embedding came from Context Vectors (CoVe) (McCann et al., 2017) and uses pre-trained word embeddings from GloVe to feed the layers of an encoder of a Seq2Seq Neural MT model which is later trained in a supervised way. A refinement of CoVe that uses a bidirectional LSTM [1] language model, called Embeddings from Language Models (ELMo) (Peters et al., 2018), was able to handle polysemy, i.e. the situation where the meaning of the words vary according to the context. The implementation of a recently proposed method for improving the performance of LSTM based language models, referred to as AWD-LSTM, resulted in ULMFiT: Universal Language Model Fine-

1. Long Short-Term Memory (LSTM) is an ANN structure usually used for sequence analysis like text, as it offers some king of memory to machines

Tuning (Howard and Ruder, 2018), that surpassed the state of the art performances in 6 text classification tasks.

The first implementation of Transformers was in a similar to ULMFiT model by replacing the LSTM structure, that led to Generative Pre-Training (GPT), (Radford et al., 2018). GPT beat the state of the art models in TC and on 8 other NLP tasks. A new implementation of Transformers that made a different use of bidirectional information than ULMFiT and GPT led to BERT (Devlin et al., 2019), which obtained new state of the art results on 11 NLP tasks, TC included.

Most of the transformer-based models for NLP tasks contain a large number of trainable parameters which are pre-trained as language models on large text corpuses like Wikipedia. These pre-trained models, like BERT, can be downloaded from the web.

3. Conformal Prediction

The typical classification problem using ML techniques, consists of building a classifier by setting a learning algorithm as a base and then training it on a data set Z where target categories are known, called training set. Then the classifier is used for predicting the category of any given instance, from a pre-specified set of categories. We define training set $Z = \{z_1, \ldots, z_n\} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$, where $n \in \mathbb{N}$ is the number of training instances, $z_i = (x_i, y_i) \in Z, \quad i \in \{1, \ldots, n\}$ is the i^{th} example, $x_i \in \mathbb{R}^d$ is a set of vector attributes and $y_i \in Y = \{Y_1, \ldots, Y_c\}$ its corresponding classification.

The CP framework, which extends conventional ML techniques, supplements the predictions of the underlying classifier with reliable confidence information, provided that the general i.i.d. assumption (data are independent and identically distributed) holds true for Z. For each new instance x_{n+1}, p-values are assigned to each possible category $Y_j \in Y$ that signify the likelihood of it being the true category. This is done by measuring the likelihood of each of the extended sets for all $Y_j \in Y$,

$$\{(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, Y_j)\}, \tag{1}$$

being drawn independently, from the same probability distribution. This is equivalent to measuring the likelihood of Y_j being the true category of instance x_{n+1}, since (x_{n+1}, Y_j) is the only new addition to the exchangeable, by assumption, set Z.

To obtain the p-value of Y_j, the strangeness of each $(x_i, y_i) \in Z$ and (x_{n+1}, Y_j) with respect to the extended set (1) is quantified by a nonconformity score with a nonconformity measure A:

$$\alpha_i^{Y_j} = A(\{(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, Y_j)\}, (x_i, y_i), \quad i = 1, \ldots, n+1. \tag{2}$$

The p-value of Y_j is calculated by comparing the nonconformity score of (x_{n+1}, Y_j) to all other nonconformity scores obtained with (2):

$$p(Y_j) = \frac{|\{i = 1, \ldots, n : \alpha_i^{Y_j} \geq \alpha_{n+1}^{Y_j}\}| + 1}{n+1}. \tag{3}$$

The final product of CP for each new instance x_{n+1}, is the *prediction-set* $\Gamma_{n+1}^{\epsilon} = \{Y_j : p(Y_j) > \epsilon\}$. Prediction-set $\Gamma_{n+1}^{\epsilon} \subseteq Y$ includes the true category with a probability $1 - \epsilon$,

where ϵ is the pre-defined significance level, (Vovk et al., 2005). This is based on the property of (3): $\forall \epsilon \in [0,1]$ and for all i.i.d. probability distributions P,

$$P^{n+1}\{p(Y_j) \leq \epsilon\} \leq \epsilon. \tag{4}$$

Because of (4), all Y_j with $p(Y_j) \leq \epsilon$ are either not the true label of instance x_{n+1} or, a case in which an event of at most ϵ probability occurred. Therefore the chance of rejecting the true category is at most ϵ.

Alternatively, if a single prediction is required the *forced prediction* can be used. Forced-prediction is composed by 3 elements. One is the most likely category for instance x_{n+1}, defined as the Y_j with the highest p-value. The other 2 are credibility which is the p-value of that Y_j and the confidence score which is defined as one minus the second highest p-value.

3.1. Inductive Conformal Prediction

The original CP also referred to as transductive CP requires the learning algorithm to be trained c times just to deal with one new instance x_{n+1}, where c is the number of possible classifications. This is a computationally heavy process and usually prohibitive when the learning algorithm is a deep ANN with millions of free parameters like Transformers.

Inductive CP for classification tasks, (Papadopoulos et al., 2002), deals with this limitation. Under this perspective the training data is split into the proper-training set $Z_{proper} = \{z_1, \ldots, z_q\}$ and the calibration set $Z_{calibration} = \{z_{q+1}, \ldots, z_n\}$ data. A general classifier is trained on the proper training set just once and the p-values are calculated by extending the calibration set with the new instance to produce the extended sets:

$$\{(x_{q+1}, y_{q+1}), \ldots, (x_n, y_n), (x_{n+1}, Y_j)\}, j = 1, \ldots, c. \tag{5}$$

The nonconformity scores of $Z_{calibration}$ are calculated just once:

$$\alpha_i = A(\{(x_1, y_1), \ldots, (x_q, y_q)\}, (x_i, y_i)), \quad i = q+1, \ldots, n. \tag{6}$$

For each new instance x_{n+1} the nonconformity scores are calculated for each possible label Y_j as:

$$\alpha_{n+1}^{Y_j} = A(\{(x_1, y_1), \ldots, (x_q, y_q)\}, (x_{n+1}, Y_j)). \tag{7}$$

The p-values of each Y_j are then:

$$p(Y_j) = \frac{|\{i = q+1, \ldots, n : \alpha_i^{Y_j} \geq \alpha_{n+1}^{Y_j}\}| + 1}{n+1}. \tag{8}$$

Given the p-values the remaining process is the same as in the original CP.

4. The Proposed Model

The proposed classifier and text prepossessing (tokenization) follows the work of Devlin et al. (2019). We use the pre-trained model BERT-base[2] and add a fully connected output layer on top for the task of text classification. We fine-tune BERT by adjusting the model

2. BERT-base can be downloaed from https://github.com/google-research/bert

parameters through re-training on new examples of in-domain text. We experiment with softmax and sigmoid activation functions. Although the sigmoid activation function is not recommended for binary or multi-class TC problems, we employed it, aiming to investigate if the additional information provided by having two independent outputs (as opposed to the 2 complimentary probabilities of softmax) can be beneficial when performing ICP.

4.1. BERT

BERT-base is trained on the BookCorpus (800M words) and English Wikipedia (2500M words) with a special language model called masked language model with two objectives:

- To predict a word within a sequence of words regardless of the direction in which the sequence is processed, in contrast with the typical language models where only the next word of a sequence can be predicted
- Next sentence prediction, where a binary classifier is used in order to determine whether two sentences come in sequence

BERT-base consists of 12 identical sequential layers referred to as transformer blocks with hidden size of 768. Each transformer block contains a self-attention head and a feed-forward layer resulting in 110 million trainable parameters in total.

BERT's input is a sequence of N tokens, where $N \leq 512$ and the output is the representation of each token. BERT uses wordpiece tokens (e.g."playing" is split into tokens: "play" and "##ing") with a 30k token vocabulary. The first token of every sequence is required to be the special classification token [CLS], while the token [SEP] is used between paired sentences and the token [PAD] is used for padding sequences of different lengths. The final output of the tokenization process is a sequence of integers corresponding to the vocabulary tokens. For words not found in the vocabulary the special token [UNK] is used.

4.2. BERT for TC

Our classifier is composed of BERT-base with a fully connected layer on top of it, as shown in Figure 2. Text is turned into tokens and then into the corresponding vocabulary ids which are the inputs of the first transformer block. Each transformer block outputs a sequence of N real valued vectors of length 768 which is the input of the next transformer block.

The final hidden state of the [CLS], which is the first element of the output sequence of the 12^{th} transformer block, is the representation (word embeddings) of the whole text and it is fed to the final layer of our classifier. That is a fully-connected layer with input size: 768 and output size: 2. To prevent over-fitting we use dropout equal to 0.1. The final output of the classifier is 2 values in $[0, 1]$ corresponding to the likelihood of each label being the true label. The classifier comes in two flavours depending on the final activation function: sigmoid and softmax. While both activation functions give output for each label in the range [0,1], softmax ensures that the sum of the probabilities for all labels equals to 1. The final sentiment is the one with the highest value.

We used the Adam optimizer (Kingma and Ba, 2014) and an early stopping criterion of accuracy with patience time set to 3 epochs. We used a learning rate of 4e-5 for the parameters of BERT and 1e-3 for the classification layer, similar to the work of Devlin

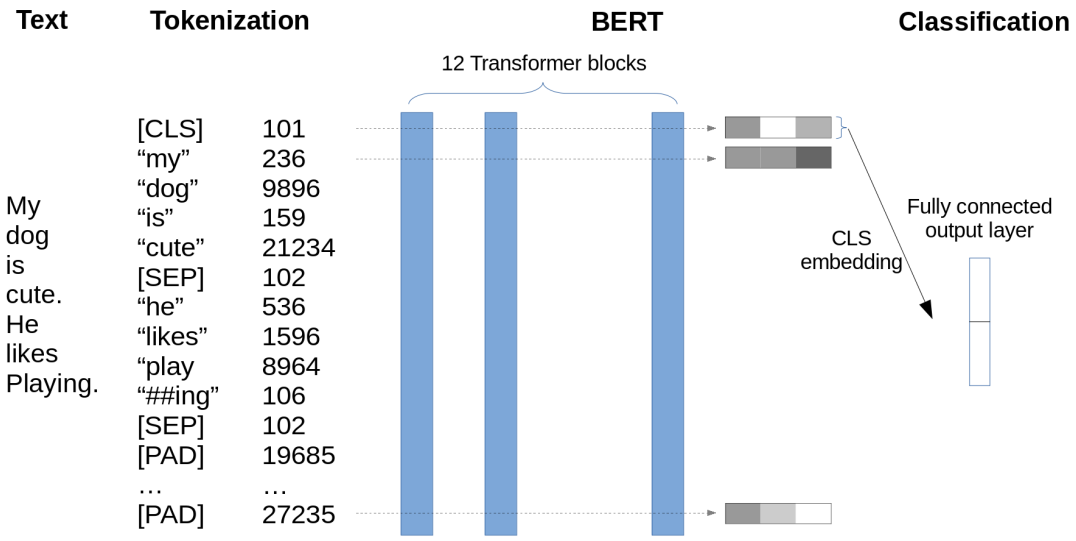

Figure 2: Architecture of the classifier model

et al. (2019) to deal with the danger of catastrophic forgetting (Howard and Ruder, 2018), which eliminates the benefit of the information captured through the pre-training phase. We also determined experimentally, that a higher learning rate for the final layer, which is randomly initialized, allows for better model convergence.

4.3. Nonconformity

We apply a typical nonconformity metric to the calibration set instances and then to each of the test instances for positive and negative label cases. The metric that we used to obtain the nonconformity scores is from the family of $p - norms$, for $p = \infty$ (also referred to as L-infinity norm):

$$L_\infty = \|\boldsymbol{c}\|_\infty = max(|c_1|, |c_2|). \tag{9}$$

The nonconformity scores for the calibration instances and for each of the test instances x_{n+1} with category Y_j are given below. The raw prediction of the classifier, that is the probability for each label being the true one, is noted as $y^\star$.

$$\begin{aligned} \alpha_i &= \|y_i^\star - y_i\|_\infty \\ \alpha_{x_{n+1}}^{Y_j} &= \|y_{n+1}^\star - Y_j\|_\infty. \end{aligned} \tag{10}$$

5. Experiments and Results

We conduct 2 experiments for each of the softmax and sigmoid classifiers. The first using the conventional model on the full training set and the second using ICP with the under-

lying model trained only on the proper training set. We compare the results of ICP with those of the corresponding conventional classifiers and investigate whether the additional information provided by sigmoid's non-complementary outputs can be beneficial.

5.1. Dataset

We use the binary IMDB movie reviews dataset (Maas et al., 2011). It contains 25k positive and 25k negative reviews and less than 30 reviews per movie. The average number of words per instance is 292. We use the first $N = 256$ words (padding size) of each instance. We split the data set into 34k training set, 6k validation set and 10k test set. For the ICP experiments we further split the training set to 23.8k proper training set (70%) and 10.2k (30%) calibration set.

5.2. Performance Measures

Experimental results are evaluated using 2 types of metrics corresponding to the 2 possible outputs of CP: forced-prediction and prediction-sets (assessing the quality of p-values). In forced prediction, we use Classification Accuracy (CA), average-confidence ($\overline{Conf.}$) and average-credibility ($\overline{Cred.}$). We define n as the total number of test instances.

- *Classification accuracy* (CA), is the average of correct predictions over the total number of test instances:

$$CA = \frac{1}{n}\sum_{i=1}^{n} I(\hat{y}_i = y_i), \tag{11}$$

  where I is 1 if the condition is true and 0 otherwise, y_i is the true category and $\hat{y}_i$ is the predicted category of the classification process for instance i.

- The confidence level indicates the likelihood of the output classification compared to the other possible classifications. We calculate *Average-confidence* ($\overline{Conf.}$) as a summary over the total number of test instances, defined as:

$$\overline{Conf.} = \frac{1}{n}\sum_{i=1}^{n} Conf_i, \tag{12}$$

  where $Conf_i$ is the confidence level of forced prediction for test case i.

- Low Credibility indicates that the particular instance is strange of all classifications. We measure *Average-credibility* ($\overline{Cred.}$), defined as:

$$\overline{Cred.} = \frac{1}{n}\sum_{i=1}^{n} Cred_i, \tag{13}$$

  where $Cred_i$ is the corresponding credibility of i test case of forced prediction.

The second type of metrics evaluates the quality of the p-values produced by ICP and the consequently the usefulness of the resulting prediction-sets. We used three of the probabilistic metrics proposed by Vovk et al. (2016), in all cases small values are preferable:

- The *s-criterion (S)*, measures the efficiency of the p-vales as their average sum across all test instances. It is defined as:

$$S = \frac{1}{n} \sum_{i=1}^{n} \sum_{y} p_i^y. \tag{14}$$

- The *of-criterion (OF)*, (OF stands for observed-fuzziness) is defined as the average sum of the p-values for the false labels. It is defined as:

$$OF = \frac{1}{n} \sum_{i=1}^{n} \sum_{y \neq y_i} p_i^y, \tag{15}$$

  where y_i is the true category.

- The *n-criterion* (N), is defined as the average prediction-size:

$$N = \frac{1}{n} \sum_{i=1}^{n} |\Gamma_i^\epsilon|. \tag{16}$$

5.3. Results

We detail experimental results in 2 parts. Section 5.3.1 evaluates forced-predictions and Section 5.3.2 prediction-sets and the quality of p-values.

5.3.1. Forced-Prediction

Table 1 reports the classification results of the underlying classifiers (without ICP), indicated by $*$, and compares them to the corresponding ICP forced-predictions. Forced-prediction performance is almost equal to the non CP-classification in both cases. This shows that the use of CP causes no substantial performance loss. The small performance loss of CP classification can be attributed to the smaller training set size of the underlying classifier, i.e. proper training set is 10.2k instances shorter. Comparing the performance of the 2 ICPs we observe that there is insignificant difference between them the 2 in all three metrics.

5.3.2. Prediction-Sets

Table 2 shows the results for the S and OF criteria. We can see that using softmax over sigmoid slightly increases both the S and OF criteria. This difference might be due to the extra information provided by the independent sigmoid outputs, however it is insignificant for any actual conclusion to be drawn.

Table 1: Forced Prediction Classification

Experiment	CA	$\overline{\text{Conf.}}$	$\overline{\text{Cred.}}$
$IMDB_{sigmoid}$	0.9198	0.9868	0.5146
$IMDB_{softmax}$	0.9137	0.9832	0.5195
$IMDB^{*}_{sigmoid}$	0.9228	-	-
$IMDB^{*}_{softmax}$	0.9202	-	-

Table 2: S & OF Criteria

Experiment	$S-Criterion$	$OF-Criterion$
$IMDB_{sigmoid}$	0.5277	0.0251
$IMDB_{softmax}$	0.5363	0.0313

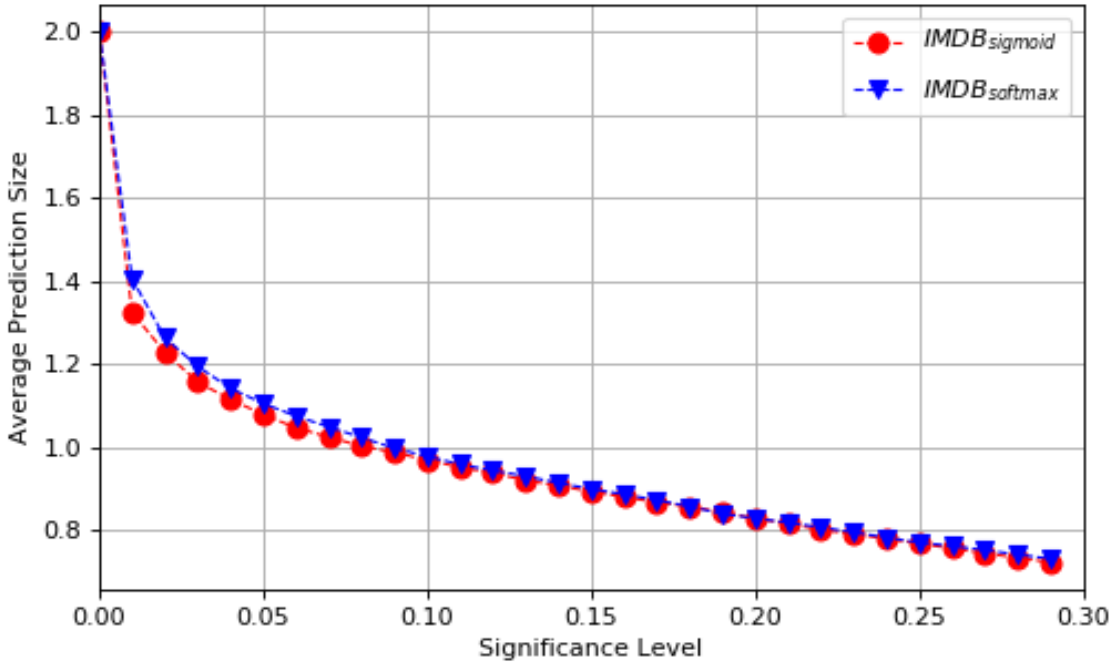

Figure 3: Average Prediction-Set Size

Average prediction-set sizes (i.e. N-criterion) for both classifiers is depicted in Figure 3 for significance levels in the range [0, 0.2]. Overall, prediction-set sizes decrease rapidly offering usable results already at relatively high confidence levels. At the confidence level of 99% (significance level $\epsilon = 1\%$), prediction-set sizes are around 1.4. At the confidence level

of 95% ($\epsilon = 5\%$), prediction-set sizes are close to 1 (on average 1.1 out of 2). At confidence level below 91% ($\epsilon \geq 9\%$) average prediction-set size falls below 1 because of empty sets.

Comparing the sigmoid and softmax versions, sigmoid classifier always offers tighter prediction-sets over softmax, but the performance difference is small. The highest difference between them is on the smallest significance level we report ($\epsilon = 1\%$), where prediction-set size is equal to 1.405 for the softmax and to 1.3246 for the sigmoid classifier.

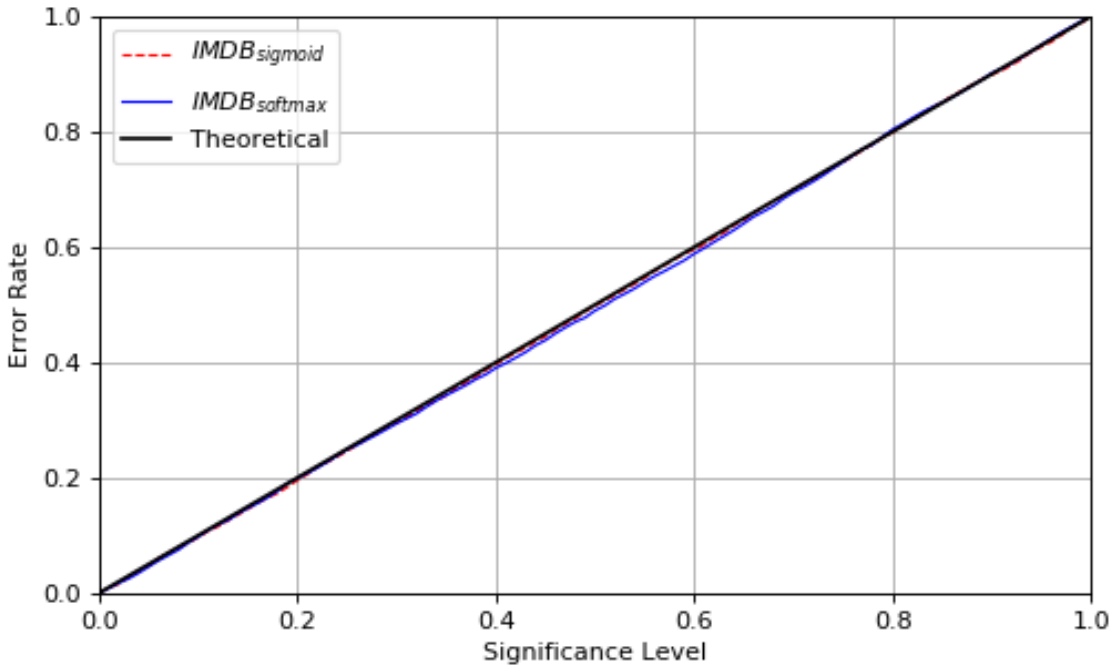

Figure 4: Empirical Error Rate

Empirical error rate results are presented in Figure 4 for significance level varying in range [0,1]. Error rate for both classifiers is almost equal and no more than the significance level, as is theoretically guaranteed by the CP framework.

6. Conclusions

We investigate the application of inductive CP combined with a transformer-based model to deal with the problem of document level SA. Specifically, our underlying classifier is a state of the art model for NLP tasks, i.e. BERT for TC. We experiment on a binary text classification problem using the IMDB movies review dataset. We also examine performance differences between softmax and sigmoid output activation functions. Our results are summarized using performance metrics for forced-predictions and prediction-sets.

The obtained results indicate that the ICP extension does not significantly affect performance in terms of classification accuracy. We marked an accuracy of 0.9228, using a simple

training strategy on a BERT model, confirming the suitability of transformer-based models for TC. Additionally, prediction-sets are overall tight enough to be useful in practise.

We intend to extend this work by experimenting with the new state of the art transformer-based models, like XLNet (Yang et al., 2019) and RoBERTa (Liu et al., 2019) to examine further performance improvements. We also plan to alter the training strategy, using guidance from the work of Sun et al. (2019), since they proved that a significant performance increase can be achieved. Furthermore, we intent to explore multi-class and multi-label versions of the problem where the additional information provided by CP can be even more useful.

References

Lambrou Antonis, Nouretdinov Ilia, and Papadopoulos Harris. Inductive venn prediction. *Annals of Mathematics and Artificial Intelligence*, 74(1):181–201, May 2015. doi: https://doi.org/10.1007/s10472-014-9420-z.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

Erik Cambria, Soujanya Poria, Alexander F. Gelbukh, and Mike Thelwall. Sentiment analysis is a big suitcase. *IEEE Intelligent Systems*, 32:74–80, 2017.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Gregory Goth. Deep or shallow, nlp is breaking out. *Communications of the ACM*, 59: 13–16, 03 2016. doi: 10.1145/2874915.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. doi: 10.18653/v1/p18-1031. URL http://dx.doi.org/10.18653/v1/p18-1031.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. Fasttext.zip: Compressing text classification models, 2016. URL http://arxiv.org/abs/1612.03651. cite arxiv:1612.03651Comment: Submitted to ICLR 2017.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014. URL http://arxiv.org/abs/1412.6980. International Conference for Learning Representations.

Bing Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2: 627–666, 2010.

Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, May 2012. doi: 10.2200/s00416ed1v01y201204hlt016.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language*

Technologies, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P11-1015.

Bryan McCann, James Bradbury, Caiming Xiong, and Richard Socher. Learned in translation: Contextualized word vectors, 2017.

Walaa Medhat, Ahmed Hassan, and Hoda Korashy. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5:1093–1113, 2014.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119. Curran Associates, Inc., 2013a.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751, Atlanta, Georgia, June 2013b. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/N13-1090.

Mika V. Mäntylä, Daniel Graziotin, and Miikka Kuutila. The evolution of sentiment analysis—a review of research topics, venues, and top cited papers. *Computer Science Review*, 27:16–32, Feb 2018. doi: 10.1016/j.cosrev.2017.10.002.

Harris Papadopoulos. Reliable probabilistic classification with neural networks. *Neurocomput.*, 107:59–68, May 2013. ISSN 0925-2312. doi: 10.1016/j.neucom.2012.07.034. URL https://doi.org/10.1016/j.neucom.2012.07.034.

Harris Papadopoulos, Vladimir Vovk, and Alex Gammerman. Qualified predictions for large data sets in the case of pattern recognition. In M. Wani, H. Arabnia, K. Cios, K. Hafeez, and G. Kendall, editors, *Proceedings of the International Conference on Machine Learning and Applications*, pages 159–163. CSREA Press, 2002. Proceedings of the International Conference on Machine Learning and Applications, CSREA Press, Las Vegas, NV, pages 159-163, 2002.

Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018. doi: 10.18653/v1/n18-1202. URL http://dx.doi.org/10.18653/v1/n18-1202.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9:371–421, 2007. URL http://arxiv.org/abs/0706.3188.

Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. How to fine-tune bert for text classification? *Chinese Computational Linguistics*, page 194–206, 2019. ISSN 1611-3349. doi: 10.1007/978-3-030-32381-3_16. URL http://dx.doi.org/10.1007/978-3-030-32381-3_16.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. Sequence to sequence learning with neural networks, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer-Verlag, Berlin, Heidelberg, 2005.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In Alexander Gammerman, Zhiyuan Luo, Jesús Vega, and Vladimir Vovk, editors, *Conformal and Probabilistic Prediction with Applications*, pages 23–39, Cham, 2016. Springer International Publishing. ISBN 978-3-319-33395-3.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Ruslan Salakhutdinov, and Quoc V. Le. Xlnet: Generalized autoregressive pretraining for language understanding, 2019.