Proceedings of Machine Learning Research 105:1–18, 2019 Conformal and Probabilistic Prediction and Applications

A Deep Neural Network Conformal Predictor for Multi-label Text Classification

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

Ladislav Lenc LLENC@KIV.ZCU.CZ
Jiří Martínek JIMAR@KIV.ZCU.CZ
Pavel Král PKRAL@KIV.ZCU.CZ
Dept. of Computer Science and Engineering, University of West Bohemia,
Plzeň, Czech Republic

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

Editor: Alex Gammerman, Vladimir Vovk, Zhiyuan Luo and Evgueni Smirnov

Abstract

We investigate the use of inductive conformal prediction (ICP) for the task of multi-label text classification and present preliminary experimental results for a subset of the original Reuters-21578 data-set. Our underlying classification model is a deep neural network configuration which consists of a trainable embedding layer, a convolutional layer and two dense feed-forward layers, arranged sequentially, with sigmoid outputs representing the individual unique labels of the selected subset. Following the power-set approach, we assign nonconformity scores to label-sets from which the corresponding p-values and prediction-sets are determined and we experiment with a number of different versions of a nonconformity measure. Our results indicate a good performance for the underlying model which is carried on to the ICP without any significant accuracy loss and with the added benefits of prediction-specific confidence information. Prediction-sets are tight enough to be practically useful even though the multi-label subset contains tens of thousands of possible label combinations and empirical error-rates confirm that our outputs are well-calibrated.

Keywords: conformal prediction, inductive conformal prediction, text classification, multi-label classification, deep neural networks, convolutional neural networks, confidence measures, Reuters-21578

1. Introduction

The task of automatic text classification is of increasing importance as the number of electronically produced and distributed texts grows, which creates a need for more efficient information storage and retrieval processes. Specifically, the problem requires that a system automatically assigns a text, e.g. a document, to a set of one or more topics, i.e. categories or classes. In practice we are dealing with either a binary scenario, where the text belongs to one of two categories; a multi-class scenario, where the text belongs to one out of many categories; or a multi-label scenario, where the text belongs to one or more categories out of a set of possible categories. Of particular interest is the last case, which frequently corresponds to real-world requirements and which is often more challenging and resource demanding than its binary and multi-class counterparts.

A first challenge involves formulating the problem in such a way that it allows for the training of the underlying machine-learning classification model and produces results that are meaningful in a multi-label setup. Approaches are grouped into two broad categories: problem transformation methods and algorithm adaptation methods, as outlined by Tsoumakas and Katakis (2007). In problem transformation methods the multi-label problem is reformulated, usually as a binary classification problem, e.g. (Gonçalves and Quaresma, 2003), or as a multi-class classification problem using all relevant category combinations, i.e. power-set approach, e.g. (Boutell et al., 2004). In the case of algorithm adaptation, solutions involve the modification of the underlying model in ways that allow it to handle multi-label outputs, e.g. neural networks in (Zhang and Zhou, 2006).

Much emphasis has also been put on the development of various feature selection and transformation methods, as it is necessary to quantify the various textual attributes that allow for the partitioning of texts into the various categories; to select the most relevant and information-rich out of the total number of attributes; and to properly prepare them as classification model inputs. However, much of current research has been oriented towards the use of artificial neural networks (ANN) that generally require minimal effort for manual feature engineering and have been shown to outperform traditional approaches such as decision trees and support vector machines (SVM).

The use of ANN models has also led to the development of more sophisticated and automated feature engineering techniques, some of which are themselves formulated as supervised learning tasks, e.g. word embedding models, and which have been shown to capture complex syntactic and semantic structures thus creating good quality features for the underlying classifiers, as in (Lilleberg et al., 2015). Classification performances are also conditioned on the quality of post-processing, as model outputs are usually probabilistic and additional analysis and expert decision-making is necessary to produce final, discrete predictions.

A significant limitation of many classification methods is that they do not provide any indication of the likelihood of their predictions being correct, and even in cases where they do provide probabilistic predictions their outputs can be misleading, as shown e.g. by Melluish et al. (2001), Papadopoulos (2013) and Lambrou et al. (2015). Conformal prediction (CP) aims to address this issue by supplementing conventional classification predictions with reliable confidence measures which are guaranteed under the assumption of data exchangeability, (Shafer and Vovk, 2007). This confidence information can be of particular

importance when dealing with classification tasks of low error tolerance, but can also be of use in a wider range of applications such as text-classification and, in general, in tasks where we are looking to pre-define specific confidence levels within which to operate. Various attempts to address issues regarding the computational efficiency of CP have been made, such as Inductive conformal prediction (ICP) which also operates under the exchangeability assumption and provides the same guarantees as CP but with reduced computational load.

The current work investigates the use of ICP as post-processing on a multi-label text classification model, realized by a deep convolutional neural network (CNN) which operates with an additional trainable embedding layer. The proposed approach is evaluated on a subset of the Reuters news-wire corpus (Lewis et al., 2004) and results are compared against the performance of the underlying model as well as against the theoretical guarantees of CP. We also experiment on the use of different nonconformity measures and show their effects on performances, particularly in the prediction-set mode. To the best of our knowledge, a similar evaluation of ICP on a well established multi-label text classification benchmark data-set does not exist, nor does the combination of ICP with a similar ANN configuration.

We are motivated from challenges faced in a business setting where the task of automatic text classification involves dealing with document categories of varying importance and urgency and therefore the provision of confidence information is vital for limiting the possibility of errors. Specifically, high confidence outputs can be handled automatically, whereas uncertain cases are identified and directed for manual classification.

The paper is structured as follows: Section 2 outlines related work on text categorization, Section 3 discusses CP and ICP and Section 4 describes our proposed approach including preprocessing steps and ANN model architecture. In Section 5 we detail our experimental set-up and present the final results, detailing performances for the forced-prediction and prediction-set modes of ICP.

2. Text Classification

Various algorithms have been implemented for the task of text categorization, for the single- and multi-label cases. More traditional approaches include decision trees, support vector machines (SVM) (e.g. Joachims (1998)) and Naive Bayes methods (e.g. Sang-Bum Kim et al. (2006)), which are usually trained on textual information that has been appropriately transformed into features, utilizing preprocessing techniques such as vector-space models and word-frequency models. The importance of feature selection is illustrated by the wealth of relevant methods developed for feeding the learning algorithm with handcrafted features and by the attempts to quantify feature "usefulness" through the development of various metrics, (Forman et al., 2003).

Recently however, there has been a shift towards the use of ANN and deep ANN architectures, that generally require less effort in terms of manual feature preparation (Nam et al., 2014). Their performance is exemplified in image classification tasks, e.g. (Krizhevsky et al., 2012), but deep ANN configurations have been shown to outperform traditional models in natural-language-processing (NLP) tasks as well, including classification tasks as in (Moraes et al., 2013).

Modern ANN-based NLP approaches can also be used as preprocessing for the creation of good quality features for the underlying classification models, e.g. word2vec models

in (Lilleberg et al., 2015). In Kim (2014) the author experimented with convolutional neural networks in the area of sentence classification and used pre-trained word embeddings (Mikolov et al., 2013) which were then compared to randomly initialized vectors. Broadly speaking, in all but a few cases, it has been shown that the usage of pre-trained word embeddings is beneficial even if they are kept static and perform better than randomly-initialized ones.

In Conneau et al. (2016), authors try to learn a high-level hierarchical representation of a sentence and at the same time operate on a low level representation of the texts, i.e. the characters. Contrary to Kim's approach, the proposed architecture is much deeper. A long short-term memory (LSTM) method is presented by Johnson and Zhang (2016), where authors use a sophisticated region embedding. Their results indicate that embeddings of text regions, which can convey complex concepts, are more useful than embeddings of single words in isolation. Peng et al. (2018) propose a graph-CNN based deep learning model to first convert texts to graph-of-words, and then use graph convolution operations to convolve the word graph and improve a large-scale hierarchical text classification success rate.

In this paper, we transform texts to numerical inputs and substitute the need for word2vec representations by placing a trainable embedding layer on-top of our deep net. The specifics are outlined in Section 4.

3. Conformal Prediction

This section discusses the concept and application of CP in the context of classification and multi-label classification problems and outlines the use of *inductive*-CP (ICP), which aims to address the computational inefficiencies of classical CP, referred to as *transductive*-CP (TCP).

The objective in a typical classification problem is to correctly predict the category to which a new, unseen example belongs. This is achieved by training a classification model on a set of known examples, i.e. training instances, of the form $\{z_1, ..., z_n\}$ where each $z_i \in Z$ is a pair (x_i, y_i), which consists of a set of attributes $x_i \in \mathbb{R}^d$ and a corresponding category $y_i \in \{Y_1, ..., Y_c\}$. A new example x_{n+1} is then assigned to a classification y_{n+1}, which is singled out from the full set of possible classifications $\{Y_1, ..., Y_c\}$, usually based on post-processing of the resulting probabilistic (or non-probabilistic) model outputs.

In the CP framework, and provided that the assumption of exchangeability holds for Z, the objective is to produce a *prediction-set* $\Gamma^{\epsilon}_{n+1} \subseteq \{Y_1, ..., Y_c\}$ that will contain the true category of the example x_{n+1} with a probability $1 - \epsilon$, for a pre-defined *significance level* ϵ (Vovk et al., 2005b). This is achieved by first assuming all possible classifications $Y_j \in \{Y_1, ..., Y_c\}$ for x_{n+1} and then determining the likelihood that each of the extended sets:

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

consists of exchangeable samples. Since (x_{n+1}, Y_j) is the only artificial pair, we are, in effect, assessing the likelihood of Y_j being the true category of x_{n+1}. This likelihood is quantified by first mapping each pair (x_i, y_i) in (1) to a numerical score using a *nonconformity measure*

A:

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

$$\alpha_{n+1}^{Y_j} = A(\{(x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, Y_j)\}, (x_{n+1}, Y_j)). \tag{2b}$$

The score $\alpha_i^{Y_j}$, called the *nonconformity score* of instance i, indicates how nonconforming, or strange, it is for z_i to belong in (1). In effect the nonconformity measure is based on a conventional machine learning algorithm, called the *underlying algorithm* of the corresponding CP and measures the degree of disagreement between the actual label y_i and the prediction $\hat{y}_i$ of the underlying algorithm, after being trained on (1). Our choice of nonconformity measure is discussed in Section 4.3.

The nonconformity score $\alpha_{n+1}^{Y_j}$ is then compared to the nonconformity scores of all other examples to find out how unusual (x_{n+1}, Y_j) is according to the nonconformity measure used. This comparison is performed with the function

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

the output of which is called the *p-value* of Y_j. An important property of (3) is that $\forall \delta \in [0, 1]$ and for all probability distributions P on $\mathcal{Z}$,

$$P^{n+1}\{((x_1, y_1), \ldots, (x_n, y_n), (x_{n+1}, y_{n+1})) : p(y_{n+1}) \leq \delta\} \leq \delta; \tag{4}$$

a proof can be found in (Vovk et al., 2005a). According to this property, if $p(Y_j)$ is under some very low threshold, say 0.05, this means that Y_j is highly unlikely as the probability of such an event is at most 5% if (1) is exchangeable. Therefore we can reject it and have at most δ chance of being wrong.

Once the p-values of all possible classifications have been produced, CP can provide two kinds of outputs:

- *Forced-prediction*, where the highest p-value classification is predicted and accompanied by a confidence score equal to one minus the second highest p-value and a credibility score equal to the p-value of the predicted classification (i.e. the highest p-value).
- *Prediction-set*, where given a predefined confidence level $1 - \delta$, the prediction set $\{Y_j : p(Y_j) > \delta\}$ is generated.

In the first case, confidence is an indication of how likely the prediction is of being correct compared to all other possible classifications, whereas credibility indicates how suitable the training set is for the particular instance. Specifically, a very low credibility value indicates that the particular instance does not seem to belong to any of the possible classifications. In the second case, the produced prediction-sets will not contain the true label of the instance with at most δ probability.

3.1. Inductive Conformal Prediction

Suppose we want to apply CP to a set of k instances with c possible classifications. In the original *transductive* setting this is achieved by re-training the underlying classification model $k \times c$ times, i.e. per instance and for each possible classification, in order to obtain the necessary p-values. This process can become computationally inefficient and often prohibitive, particularly for large data-sets and for resource-demanding underlying models (such as deep ANNs).

An alternative to the above is *inductive* conformal prediction (ICP), first proposed by Papadopoulos et al. (2002a) and Papadopoulos et al. (2002b) for regression and classification tasks, respectively. In the ICP setting, the underlying model is trained only once thus generating a single general "rule" which is then applied on each test instance.

Specifically, the training data-set is first split into two components, the *proper-training set* $\{z_1, \ldots, z_q\}$ and the *calibration set* $\{z_{q+1}, \ldots, z_n\}$. The underlying model is trained on the proper-training set and the resulting model is used for calculating the nonconformity scores of the calibration instances as:

$$\alpha_i = A(\{(x_1, y_1), \ldots, (x_q, y_q)\}, (x_i, y_i)), \quad i = q+1, \ldots, n. \tag{5}$$

The trained model and the calibration set's nonconformity scores form the general "rule" of the ICP. Then, the nonconformity score for each assumed class Y_j of every test instance x_{n+m} is calculated in the same way:

$$\alpha_{n+m}^{Y_j} = A(\{(x_1, y_1), \ldots, (x_q, y_q)\}, (x_{n+m}, Y_j)), \tag{6}$$

and is used together with the nonconformity scores of the calibration instances to calculate the p-value:

$$p(Y_j) = \frac{|\{i = q+1, \ldots, n : \alpha_i \geq \alpha_{n+m}^{Y_j}\}| + 1}{n - q + 1}. \tag{7}$$

3.2. CP for Multi-label Classification

In multi-label classification problems, the true class may be a combination of various single-classes, e.g. a news-wire might be labeled under both *politics* and *oil-price*, at the same time. In such a setting, we need to consider a number of different class combinations, with a maximum limit set by the total number of all possible class combinations.

Different versions of CP have been proposed for handling multi-label classification problems through the use of problem reformulations. These include the following approaches: *power-set* (Papadopoulos, 2014), *binary-relevance* (Lambrou and Papadopoulos, 2016) and *instance reproduction* (Wang et al., 2015), and are similar in principle to the problem transformation methods used when dealing with conventional classification models which also cannot handle multi-label problems, e.g. SVM.

In this work we use the power-set approach, in which each possible *label-set* $\Psi_m \in \mathcal{P}(\{Y_1, \ldots, Y_c\})$ is treated as a candidate classification and is assigned a p-value. However, as the number of possible classifications in the particular case is extremely large, we consider only label combinations up to the maximum observed label cardinality of the corpus, as

detailed in Section 5.2. For the rest of this paper we denote this reduced set of label-sets as $\{\Psi_1, \ldots, \Psi_g\}$.

4. The Proposed Approach

4.1. Text Preprocessing

We use a three-stage preprocessing pipeline to transform the raw news-wire texts and their corresponding class-labels to proper training/test inputs and targets that can be fed to our ANN configuration and that can also be used for calculating the ICP-related variables. We firstly create a bag-of-words (i.e. vocabulary) using the N most frequent words from our training examples (Section 4.1.1). Then, all texts are transformed to numerical vectors where numbers correspond to vocabulary words (Section 4.1.2) and finally, targets are transformed to multi-hot representations (Section 4.1.3).

4.1.1. Vocabulary Creation

Vocabulary creation is based on all training instances, i.e. proper training + calibration. Digits and punctuation are removed as well as words with length smaller than a predefined minimum limit, WL_{min}. The frequency of the remaining words in the training corpus is determined, sorted in a descending order and the first N words are selected. Each word is then given a unique identification number, W_{id}, which corresponds to its vocabulary index. The experimental results presented in this paper were calculated with $WL_{min} = 3$ and $N = 20000$.

4.1.2. Texts-to-Vectors

All instances in the training and test sets are transformed into word-vectors, by substituting each instance-word by its corresponding W_{id}. Words not found in the vocabulary are replaced by a reserved identification number, i.e. $max(W_{id}) + 1$. Furthermore, in order to produce word-vectors of consistent sizes, we limit the total number of words per instance to WV_{max}. Instances with fewer words are padded to the desired size using a reserved identification number, i.e. $max(W_{id}) + 2$. Instances with more words than the predefined limit are simply cut to fit WV_{max}. For the purposes of this study WV_{max} was set to 200.

4.1.3. Targets-to-Multi-Hot

In order to produce valid targets for the ANN architecture, the label-set of each training/test instance is transformed into a multi-hot representation. Specifically, we associate each instance (x_i, ψ_i), where $\psi_i \subseteq \{Y_1, \ldots, Y_c\}$, with a 1D binary label-vector $< t_i^1, \ldots, t_i^c >$, where $t_i^j = 1$ iff $Y_j \in \psi_i$ and $t_i^j = 0$ otherwise. An example is shown in Figure 1.

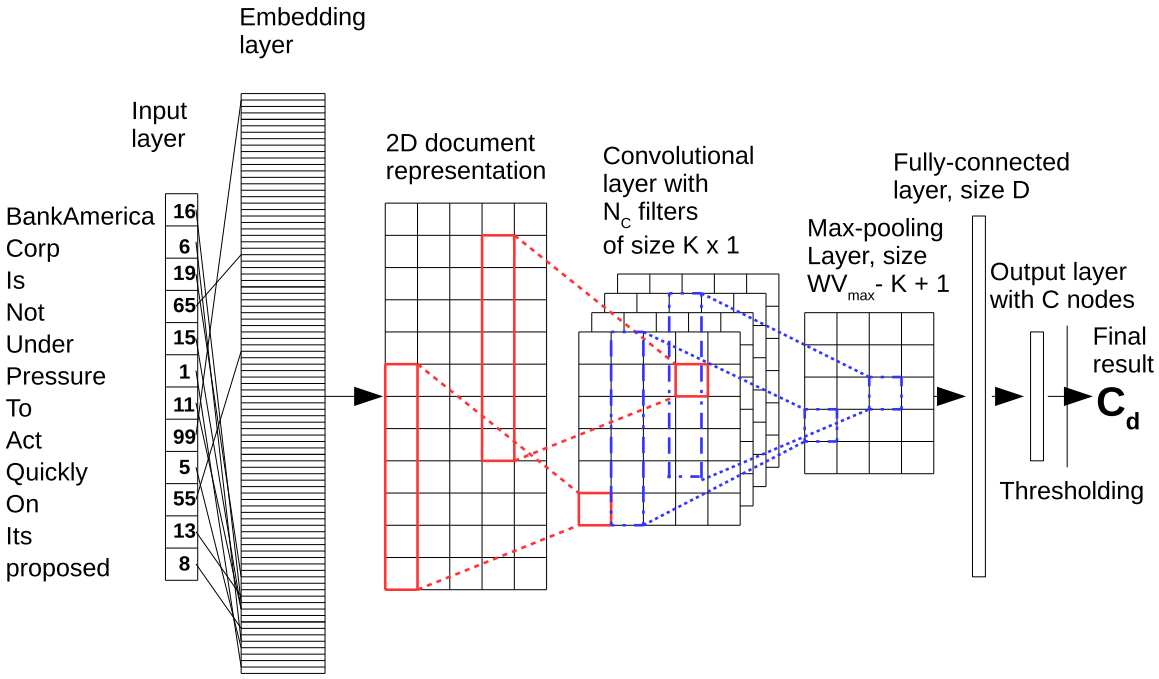
Figure 1: Multi-hot representation for the label-set $\{Y_3, Y_8\}$

0	0	1	0	0	0	0	1	0	0	0	0	0	0	0

4.2. ANN Configuration and Training

In this work, we utilize a deep ANN configuration proposed by Lenc and Král (2016) and inspired by the work of Kim (2014), which is depicted in Figure 2. Neural networks utilized in the NLP domain often rely on word embeddings that are trained on very large corpora Mikolov et al. (2013). It was proved that such embeddings are very useful especially in the case of short texts Kim (2014). However, for longer texts, better results were achieved using embeddings initialized randomly and trained jointly with the classification network Lenc and Král (2016). We thus used randomly initialized embeddings.

Figure 2: Architecture of the utilized network



Network inputs consist of word sequences of texts, represented by vocabulary indices, as described in Section 4.1. Inputs are fed to a trainable embedding layer with randomly initialized weights, which transforms them to multi-dimensional word vectors.

The embedding outputs form 2-D representations of texts which are then fed to a convolutional layer. The convolutional layer comprises of N_C filters of size $K \times 1$, where N_C and K were set to 40 and 16, respectively. The number and size of filters were chosen according to Lenc and Král (2016) where it was proved that further increasing of these values doesn't bring any significant improvement. Next is a max-pooling layer which includes a dropout option, at a rate of 0.2. The max-pooling layer reduces dimensionality and prepares the outputs for a sequence of two fully-connected layers, the first one with 256 output neurons and a dropout of 0.2 and the final one with output neurons that match the number of single-class labels (i.e. 20 for our experiments). All layers except the output one use rectified linear unit (ReLU) non-linearity.

We use sigmoid activation for outputs and the results are thresholded with a probability threshold $r_{thr} = 0.5$ (which is however only applied in the standalone mode, i.e. when CP is used the threshold is removed so as not to alter the nonconformity scores). We use binary cross-entropy (de Boer et al., 2005) as a loss function, and Adam (Kingma and Ba, 2014) as an optimizer. For all experiments, models were trained for a duration of 30 epochs.

4.3. Nonconformity Measures

We compute *nonconformity scores* for all calibration-set instances and then for each test-set instance and for each possible label-set, as described in Section 3. We experiment with four different *nonconformity measures*, all of which are derived from:

$$a_i = \sum_{j=1}^{c} |t_i^j - o_i^j|^\phi + \lambda \sum_{1 \le j < r \le c} t_i^j t_i^r \mu_{j,r}, \tag{8}$$

where c is the total number of single-class labels, t_i^j is 1 if j is a true label of instance i and 0 otherwise, o is the probabilistic output of the underlying model for instance i and label j. Parameter ϕ is used for controlling the sensitivity of the nonconformity measure, since larger ϕ values will result in smaller $|t_i^j - o_i^j|$ for smaller prediction errors. The second part of (8) penalizes label-sets which include label-pairs that have never been observed in the proper-training set, where $\mu_{j,r}$ is 1 if labels Y_j and Y_r have not been observed together and 0 otherwise. Based on (8) we construct four different versions of the nonconformity measure:

(a) with $\phi = 1$ and $\lambda = 0$

(b) with $\phi = 1$ and $\lambda = 1$

(c) with $\phi = 2$ and $\lambda = 1$

(d) with $\phi = 4$ and $\lambda = 1$

5. Experiments and Results

We conduct experiments on a subset of the Reuters news-wire benchmark data-set, described in Section 5.1. Raw texts are first partitioned into the appropriate training and test sets and the preprocessing steps outlined in Section 4.1 are applied. The underlying ANN model (Section 4.2) is trained once per experiment on the proper-training set and then deployed accordingly on the calibration and test sets. Nonconformity measures are calculated once per experiment on the calibration instances and multi-label target p-values once per test instance, per experiment, for all possible target combinations up to the maximum observed multi-label cardinality, as defined in Section 5.2.

Performances are calculated using a number of different evaluation metrics (Section 5.3) and are summarized in Section 5.4. In particular, we present results for the forced-prediction (Section 5.4.1) and prediction-set (Section 5.4.2) modes and include results for the quality of p-values and the empirical validity of our conformal predictor.

5.1. Data-sets Used

Multiple versions of the original Reuters news-wire benchmark data-set exist. For the purposes of this study we use our own subset which is however based on the ModApte (R90) split, as retrieved directly from Python's Natural-Language-Toolkit (NLTK) library (Bird et al., 2009). This version consists of 10788 examples, 7769 for training and 3019 for testing, associated with a total of 90 classes, where each class is represented at least once in the training and in the test sets.

However, the proposed multi-label ICP algorithm requires the evaluation of nonconformity measures on class label-sets, i.e. label combinations. The total number of all possible label combinations for R90 is 2^{90}, which is computationally prohibitive for our hardware setup. We therefore reduce the total number of classes and produce a new mode of the NLTK-version, which consists of the 20 most populous classes and a total number of 9266 texts (6701 for training and 2565 for testing). We refer to this as R20.

Finally, and as outlined in Section 3, ICP requires that the training set is further split into the proper-training and calibration sets. The ratio has been set to 0.7 - 0.3, i.e. 70% for proper-training and 30% for calibration, or 4691 and 2010 instances, respectively.

5.2. Label-sets Used

We further reduce computational requirements by evaluating nonconformity measures for label combinations which are subsets of all possible combinations, but only up to the maximum observed label cardinality, as determined from the complete subset (i.e. training + test sets). For R20, the maximum number of labels for a single instance was found to be 5 and we therefore discard all label-sets with more than 5 labels. This results in a total number of 21699 possible label-sets.

5.3. Evaluation Metrics

We group evaluation metrics into two categories. The first relates to the performance of the forced-prediction mode which enables the comparison of ICP with the underlying model, while the second category concerns prediction-sets and the quality of p-values.

In all cases, $\psi_i \subseteq \{Y_1, \ldots, Y_c\}$ corresponds to the true label-set and $\hat{\psi}_i$ to the predicted label-set for test instance $i \in \{1, \ldots, k\}$, while $< t_i^1, \ldots, t_i^c >$ and $< \hat{t}_i^1, \ldots, \hat{t}_i^c >$ are the multi-hot representations of ψ_i and $\hat{\psi}_i$ respectively.

Forced-prediction results are evaluated using six different metrics:

- *Classification accuracy* (CA) is computed for the complete set of test instances and averaged over their total number. For each instance, a correct prediction is given if and only if the true multi-label target has been fully matched by the highest p-value prediction, i.e.

$$CA = \frac{1}{k} \sum_{i=1}^{k} I(\psi_i = \hat{\psi}_i), \tag{9}$$

where I is 1 if the condition is true and 0 otherwise. Accuracy is therefore strict and rewards only *absolutely-correct* predictions and not *partially-correct* predictions.

- The F_1-*measure* corresponds to the harmonic mean of precision and recall and its value is in the range [0, 1]. It can be further split into the micro-averaged and macro-averaged types. F_{micro} is averaged over the complete set of test instances, which means that more frequent labels weight more than infrequent ones. Conversely, F_{macro} is first averaged per label and the results are then averaged over the total number of labels. Consequently, F_{macro} gives equal weights to all labels and it therefore tends to be lower than F_{micro} when poorer performance is observed for the more infrequent ones. The two are defined as:

$$F_{micro} = \frac{2\sum_{j=1}^{c}\sum_{i=1}^{k} t_i^j \hat{t_i^j}}{\sum_{j=1}^{c}\sum_{i=1}^{k} t_i^j + \sum_{j=1}^{c}\sum_{i=1}^{k} \hat{t_i^j}}, \tag{10}$$

$$F_{macro} = \frac{1}{c}\sum_{j=1}^{c} \frac{2\sum_{i=1}^{k} t_i^j \hat{t_i^j}}{\sum_{i=1}^{k} t_i^j + \sum_{i=1}^{k} \hat{t_i^j}}, \tag{11}$$

- *Hamming Loss (HL)* is evaluated as a loss function and, in contrast to accuracy and F_1 measures, the objective is minimization. It is given by the symmetric difference between actual and predicted labels, averaged over the total number of test instances and it is more suitable than CA for multi-label classification problems as it also rewards *partially-correct* predictions. It is defined as:

$$HL = \frac{1}{kc}\sum_{i=1}^{k}\sum_{j=1}^{c} t_i^j \oplus \hat{t_i^j}, \tag{12}$$

where $\oplus$ is the xor operator.

- We also evaluate *average-confidence* ($\overline{Conf.}$), which is intended as an overall indication of how likely predictions are compared to all other possible classifications (Section 3) and it is defined as:

$$\overline{Conf.} = \frac{1}{k}\sum_{i=1}^{k} 1 - \max_{\Psi \neq \arg\max_{\Psi} p_i(\Psi)} p_i(\Psi), \tag{13}$$

where we compute the average value of all confidence scores (i.e. $1-$ the second largest p-value p, over all considered label-sets Ψ) over k number of test instances.

- As discussed in Section 3, credibility indicates how suitable is the training data-set for each test instance. Here we evaluate an overall model suitability using *average-credibility* ($\overline{Cred.}$), defined as:

$$\overline{Cred.} = \frac{1}{k}\sum_{i=1}^{k}\max_{\Psi} p_i(\Psi), \tag{14}$$

where the credibility of example i is the largest p-value p out of all considered label-sets Ψ.

The quality of the generated p-values and the practical usefulness of the prediction-sets are evaluated using the following criteria, as proposed in (Vovk et al., 2016):

- The *s-criterion (S)*, i.e. sum criterion, is an efficiency measure given as the average sum of p-values across all test instances and it is independent of significance level ϵ. Small values are preferable. It is defined as:

$$S = \frac{1}{k}\sum_{i=1}^{k}\sum_{\Psi} p_i(\Psi) \tag{15}$$

- The *of-criterion (OF)*, i.e. observed-fuzziness criterion, is the same as the *s-criterion*, but excluding the p-value of the true label-set. It therefore evaluates the average prediction-set size for all false predictions. Smaller values are, again, preferable. It is defined as:

$$OF = \frac{1}{k}\sum_{i=1}^{k}\sum_{\Psi\neq\psi_i} p_i(\Psi) \tag{16}$$

where we sum all p-values p excluding the p-value of the true label-set.

- The *n-criterion*, i.e. number criterion, which returns the average size of the resulting predictions-sets:

$$N = \frac{1}{k}\sum_{i=1}^{k}|\Gamma_i^{\epsilon}|. \tag{17}$$

where $|\Gamma_i^{\epsilon}|$ is the size of the resulting prediction-set for instance i at a significance level ϵ.

5.4. Results

We present forced-prediction results in Section 5.4.1 and prediction-set results in Section 5.4.2. Where appropriate, we present results for all four versions of the nonconformity measure (i.e. a-d), as discussed in Section 4.3.

5.4.1. Forced Predictions

Forced-prediction results are based on the label-sets with the highest p-value, per test instance (Section 3) and are shown in Table 1. For comparison purposes, in addition to the results for nonconformity measures a-d, we also present result for the underlying model (i.e. without CP), indicated by $*$.

Table 1: Forced Prediction Results

Experiment	CA	F_{micro}	F_{macro}	HL	$\overline{\text{Conf.}}$	$\overline{\text{Cred.}}$
R20*	0.892	0.935	0.824	0.007	-	-
R20^a	0.897	0.932	0.832	0.008	0.919	0.532
R20^b	0.897	0.932	0.832	0.008	0.919	0.532
R20^c	0.897	0.932	0.832	0.008	0.939	0.533
R20^d	0.897	0.932	0.822	0.008	0.948	0.533

In contrast with R20^{a,b,c,d}, the R20* model was trained on the full set of training instances, i.e. proper training and calibration sets. Nevertheless, it can be seen that the results are approximately equal for all metrics among the five experiments presented. This indicates that no substantial classification performance is sacrificed by the use of CP, which provides important additional information with each prediction.

For all experiments, F_{macro} is lower than F_{micro} which suggests that the classification model has poorer performance for the less frequent classes. Classification accuracy is high (≈ 0.9 in all cases), and Hamming loss remains low between $0.007 - 0.008$. $\overline{\text{Conf.}}$ is for all R20^{a,b,c,d} above 0.9 which shows high certainty in the majority of predicted label-sets. Finally the average credibility is a bit higher than 0.5 (which is the expected average credibility for the true labels) and shows that the trained model is suitable for classifying the test instances. It should be noted that this relatively good performance for a multi-label classification task can be, at least partly, attributed to the fact that the large majority of the instances belong to a single class ($\approx 90\%$).

5.4.2. Prediction Sets

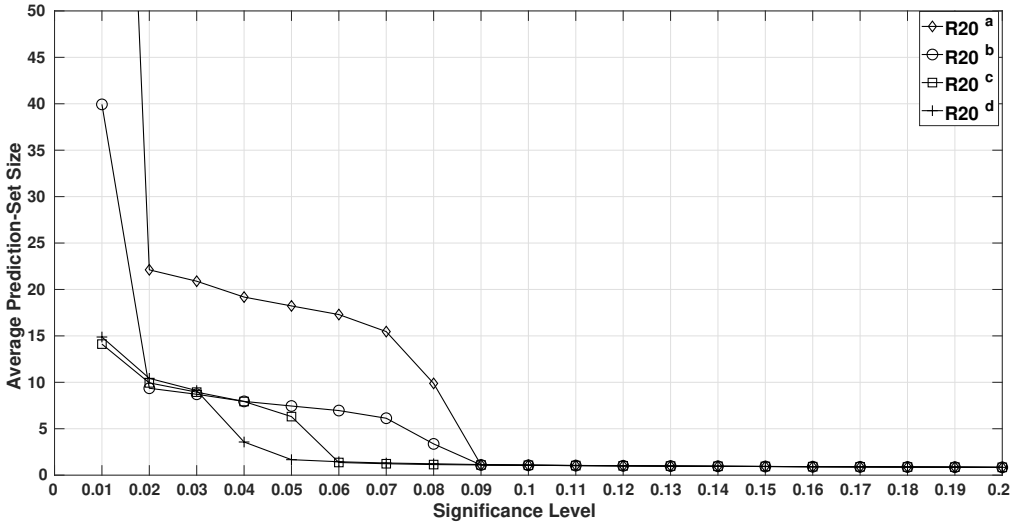
Results for the S and OF criteria are presented in Table 2. We notice a substantial decrease for both metrics between R20^a and the rest of the experiments, which is explained by the change of λ (Section 4.3) from 0 to 1. A small decrease for both values is also shown between R20^b and R20^c and between R20^c and R20^d, although it is clear that the increase in ϕ (Section 4.3) has much less impact on the two metrics compared to a change in λ.

Table 2: S & OF Criteria

Experiment	$S-Criterion$	$OF-Criterion$
R20^a	18.602	18.082
R20^b	12.928	12.409
R20^c	12.654	12.134
R20^d	12.551	12.031

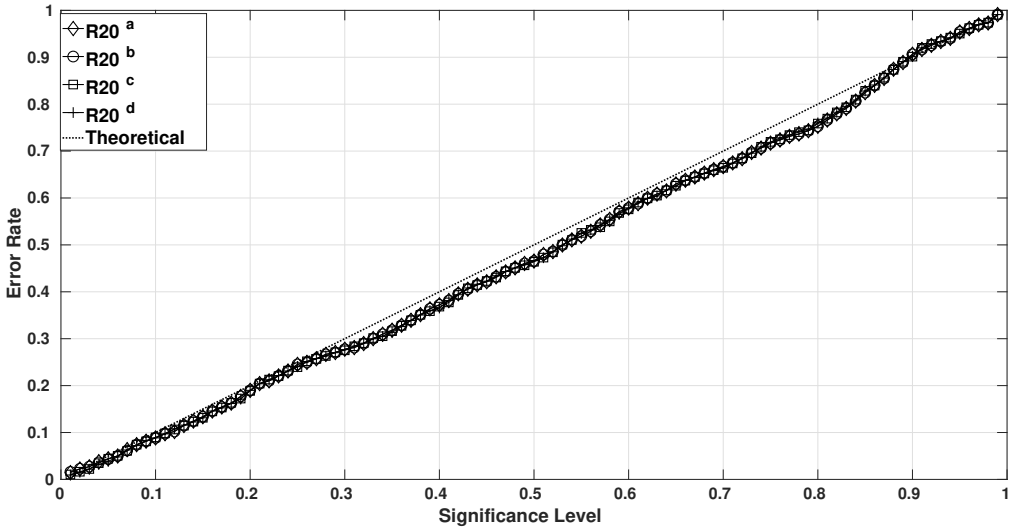
Figure 3 shows the average prediction-set sizes (i.e. n-criterion), for all four versions of the nonconformity measure and for significance levels in the range [0.01, 0.2]. Here, apart from the obvious change due to the parameter λ, we can also observe more clearly the effects of increasing the sensitivity factor ϕ, where in each step (between b-d) there are reductions in the average prediction-set sizes, especially for significance levels > 0.03. Overall the prediction sets produced by the proposed approach become very tight with R20^d providing prediction sets containing on average ≈ 15 out of the possible 21699 label-sets at the 99% confidence level, while this is reduced to only ≈ 2 label-sets at the 95% confidence level. In general for all experiments, the prediction sets stabilize to a size=1 for significance levels > 0.09.

Figure 3: Average Prediction-Set Size



We also examine the empirical validity of our conformal predictor by plotting the test-set error-rate against the significance level in the range [0, 1], presented in Figure 4. It is shown that, as guaranteed theoretically, the error-rate is always less than or equal to the significance level (up to statistical fluctuations) and confirms that our data satisfies the exchangeability assumption, even though no data randomization was used. Additionally, we show only negligible differences between all R20^{a,b,c,d} in terms of the empirical error rate.

Figure 4: Empirical Error Rate



6. Conclusions

We examined the application of the CP framework on a multi-label text classification problem and assessed its performance on a subset of the Reuters news-wire data-set. Specifically, the proposed approach follows the inductive version of the CP framework combined with a deep ANN configuration including an additional trainable embedding layer. Nonconformity scores are assigned to the possible label-sets in a power-set manner using four versions of a multi-label nonconformity measure.

Our experimental comparison between the performance of the proposed ICP in the forced prediction mode and that of its underlying model shows that ICP has a negligible negative impact on performance for providing important confidence information. Additionally, we can conclude that our underlying model performs well without the need of computationally demanding pre-trained word-embeddings.

Furthermore, the prediction sets produced by the proposed approach are well-calibrated and tight even for high confidence levels (> 0.95), especially with the last set of nonconformity measure parameter values (d), even though the number of candidate label-sets was more than 20k. This shows that the ICP prediction sets can be practically useful.

Our future plans include the evaluation of the proposed approach on non-English text corpora as well as on data-sets with higher average label-set cardinality. We are also interested in investigating the use of CP with other deep ANN configurations (such as attention-based and RNN/LSTM) and to further experiment on the effects of using different nonconformity measures.

References

Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python*. O'Reilly Media Inc., 2009.

Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. Very deep convolutional networks for text classification. *arXiv preprint arXiv:1606.01781*, 2016.

Pieter-Tjerk de Boer, Dirk P. Kroese, Shie Mannor, and Reuven Y. Rubinstein. A tutorial on the cross-entropy method. *Annals of Operations Research*, 134(1):19–67, Feb 2005. ISSN 1572-9338. doi: 10.1007/s10479-005-5724-z. URL https://doi.org/10.1007/s10479-005-5724-z.

George Forman, Isabelle Guyon, and Andr Elisseeff. An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305, 2003.

Teresa Gonçalves and Paulo Quaresma. A preliminary approach to the multilabel classification problem of portuguese juridical documents. In *Portuguese Conference on Artificial Intelligence*, pages 435–444. Springer, 2003.

Thorsten Joachims. Text categorization with support vector machines: Learning with many relevant features. In Claire Nédellec and Céline Rouveirol, editors, *Machine Learning: ECML-98*, pages 137–142, Berlin, Heidelberg, 1998. Springer Berlin Heidelberg.

Rie Johnson and Tong Zhang. Supervised and semi-supervised text categorization using LSTM for region embeddings. *arXiv preprint arXiv:1602.02373*, 2016.

Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, 12 2014.

Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012. URL http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

Antonis Lambrou and Harris Papadopoulos. Binary relevance multi-label conformal predictor. In Alexander Gammerman, Zhiyuan Luo, Jesús Vega, and Vladimir Vovk, editors, *Conformal and Probabilistic Prediction with Applications*, pages 90–104, Cham, 2016. Springer International Publishing.

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

Ladislav Lenc and Pavel Král. Deep neural networks for Czech multi-label document classification. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 460–471. Springer, 2016.

David D Lewis, Yiming Yang, Tony G Rose, and Fan Li. Rcv1: A new benchmark collection for text categorization research. *Journal of machine learning research*, 5(Apr):361–397, 2004.

J. Lilleberg, Y. Zhu, and Y. Zhang. Support vector machines and word2vec for text classification with semantic features. In *2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC)*, pages 136–140, July 2015. doi: 10.1109/ICCI-CC.2015.7259377.

Thomas Melluish, Craig Saunders, Ilia Nouretdinov, and Volodya Vovk. Comparing the Bayes and Typicalness frameworks. In *Proceedings of the 12th European Conference on Machine Learning (ECML'01)*, volume 2167 of *Lecture Notes in Computer Science*, pages 360–371. Springer, 2001.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.

Rodrigo Moraes, Joao Valiati, and Wilson Gavio Neto. Document-level sentiment classification: An empirical comparison between SVM and ANN. *Expert Systems with Applications*, 40:621633, 02 2013. doi: 10.1016/j.eswa.2012.07.059.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. Large-scale multi-label text classification revisiting neural networks. In *Joint european conference on machine learning and knowledge discovery in databases*, pages 437–452. Springer, 2014.

Harris Papadopoulos. Reliable probabilistic classification with neural networks. *Neurocomputing*, 107:59 – 68, 2013. ISSN 0925-2312. doi: https://doi.org/10.1016/j.neucom.2012.07.034. URL http://www.sciencedirect.com/science/article/pii/S0925231212007801. Timely Neural Networks Applications in Engineering.

Harris Papadopoulos. A cross-conformal predictor for multi-label classification. In Lazaros Iliadis, Ilias Maglogiannis, Harris Papadopoulos, Spyros Sioutas, and Christos Makris, editors, *Artificial Intelligence Applications and Innovations*, pages 241–250, Berlin, Heidelberg, 2014. Springer Berlin Heidelberg.

Harris Papadopoulos, Kostas Proedrou, Volodya Vovk, and Alex Gammerman. Inductive confidence machines for regression. In *Proceedings of the 13th European Conference on Machine Learning (ECML'02)*, volume 2430 of *LNCS*, pages 345–356. Springer, 2002a.

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, 2002b. Proceedings of the International Conference on Machine Learning and Applications, CSREA Press, Las Vegas, NV, pages 159-163, 2002.

Hao Peng, Jianxin Li, Yu He, Yaopeng Liu, Mengjiao Bao, Lihong Wang, Yangqiu Song, and Qiang Yang. Large-scale hierarchical text classification with recursively regularized

deep graph-cnn. In *Proceedings of the 2018 World Wide Web Conference on World Wide Web*, pages 1063–1072. International World Wide Web Conferences Steering Committee, 2018.

Sang-Bum Kim, Kyoung-Soo Han, Hae-Chang Rim, and Sung Hyon Myaeng. Some effective techniques for Naive Bayes text classification. *IEEE Transactions on Knowledge and Data Engineering*, 18(11):1457–1466, Nov 2006. ISSN 1041-4347. doi: 10.1109/TKDE.2006.180.

Glenn Shafer and Vladimir Vovk. A tutorial on conformal prediction. *CoRR*, abs/0706.3188, 2007. URL http://arxiv.org/abs/0706.3188.

Grigorios Tsoumakas and Ioannis Katakis. Multi-label classification: An overview. *Int J Data Warehousing and Mining*, 2007:1–13, 2007.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. Springer, New York, 2005a.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic Learning in a Random World*. 01 2005b. doi: 10.1007/b106715.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. *CoRR*, abs/1603.04416, 2016. URL http://arxiv.org/abs/1603.04416.

Huazhen Wang, Xin Liu, Ilia Nouretdinov, and Zhiyuan Luo. A comparison of three implementations of multi-label conformal prediction. In Alexander Gammerman, Vladimir Vovk, and Harris Papadopoulos, editors, *Statistical Learning and Data Sciences*, pages 241–250, Cham, 2015. Springer International Publishing.

Min-Ling Zhang and Zhi-Hua Zhou. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351, 2006.