Proceedings of Machine Learning Research 152:1–23, 2021 Conformal and Probabilistic Prediction and Applications

Transformer-based Conformal Predictors for Paraphrase Detection

Patrizio Giovannotti PATRIZIO.GIOVANNOTTI.2019@LIVE.RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK
Alex Gammerman A.GAMMERMAN@RHUL.AC.UK
Royal Holloway, University of London, Egham, Surrey, UK

Editor: Lars Carlsson, Zhiyuan Luo, Giovanni Cherubin and Khuong An Nguyen

Abstract

Transformer architectures have established themselves as the state-of-the-art in many areas of natural language processing (NLP), including paraphrase detection (PD). However, they do not include a confidence estimation for each prediction and, in many cases, the applied models are poorly calibrated. These features are essential for numerous real-world applications. For example, in those cases when PD is used for sensitive tasks, like plagiarism detection, hate speech recognition or in medical NLP, mistakes might be very costly. In this work we build several variants of transformer-based conformal predictors and study their behaviour on a standard PD dataset. We show that our models are able to produce *valid* predictions while retaining the accuracy of the original transformer-based models. The proposed technique can be extended to many more NLP problems that are currently being investigated.

Keywords: Conformal prediction, natural language understanding, paraphrase detection, transformers.

1. Introduction

The objective of paraphrase detection is to recognise if two different sentences are semantically equivalent. Specifically, given two word sequences $a = (a_1, \dots, a_m), b = (b_1, \dots, b_n)$, we say b is a paraphrase of a if $b \neq a$ and b conveys the same meaning of a. Paraphrase detection (PD) is therefore a quite ill-defined problem, since even the term “meaning” defies any attempt of formal, general definition. It is, nonetheless, an exciting problem as it is closely linked to the concept of “understanding” natural language and plays an important role in many downstream NLP tasks, such as text summarization, plagiarism detection and duplicate detection. PD can be also used as a supportive task, such as data augmentation for dialogue systems (Falke et al., 2020). It is an active area of research with new and challenging datasets being frequently released (see for example Zhang et al., 2019, Yang et al., 2019, He et al., 2020).

Transformers (Vaswani et al., 2017) have achieved state-of-the-art performance on several PD datasets (e.g., GLUE and its leaderboard, Wang et al., 2019) and have become the *de facto* standard for text classification tasks. Their versatile architecture, primarily based on the attention mechanism (Bahdanau et al., 2015), is fairly easy to parallelize and can be altered to suit different training strategies. BERT (Bidirectional Encoder Representations from Transformers, Devlin et al., 2019) is arguably the most notable example of applying a training strategy to a transformer. The resulting model, released by Google, can be retrained on different downstream tasks (such as PD) without the need to change the model's architecture.

While BERT and the other transformer-based models can achieve remarkable predictive performance, they fail to provide reliable estimates of confidence for their predictions. The raw output of a transformer, usually a vector of real numbers called *logits*, gives no indication about the probability of the prediction being correct. To circumvent this limitation, researchers often apply a *softmax* function that normalizes the output into a vector of numbers that add up to 1. While such an output resembles a probability distribution, there is no real connection between the softmax value and the actual, empirical probability of a predicted label $\hat{y}$ to be the true label. This becomes a significant limitation with some "sensitive" use cases: for example, a social network may want to automatically remove potential hate speech contents only when the model's confidence is higher than a certain threshold; the same social network may want to battle fake news by finding all the different posts that tell the same story with high probability; an academic institution may want to investigate cases of plagiarism that are highly likely to be confirmed and so on.

We propose a range of predictors that are able to produce *valid* confidence estimates for each test example and preserve the predictive power of Transformers. Our proposed models are based on conformal prediction (CP, Vovk et al., 2005), a machine learning framework that can be built on top of any ML algorithm. The property of validity implies that the error rate ϵ of a predictor can be controlled in advance, a property that CP guarantees with the only assumption of the examples being exchangeable. We show that by using a fine-tuned BERT model as underlying algorithm we are able to build a valid conformal predictor and we study which CP variant offer better performances for paraphrase detection.

The main contributions of our work are the following:

- We describe a method of reliable uncertainty estimation for the paraphrase detection task that requires minimal assumptions
- We experiment with several variants of conformal predictors like Mondrian and cross-conformal predictors
- We introduce a nonconformity measure based on the transformer's raw output scores that does not rely on a softmax or logistic function.

2. Related work

Maltoudoglou et al. (2020) described a BERT-based conformal predictor applied to sentiment classification. The authors trained an inductive conformal predictor on the IMDB movie reviews dataset with a nonconformity measure based on BERT's output. They showed how the resulting model was valid and retained BERT's original predictive performance. Paisios et al. (2019) applied conformal prediction to a multi-label text classification task.

The theme of confidence estimation in NLP (Gandrabur et al., 2006) is attracting growing interest from both academia and industry, with most studies focused on Bayesian approaches. Recently Shelmanov et al. (2021) described a dropout-based uncertainty estimation technique applied to Transformer-based models. In machine translation, the first techniques (Blatz et al., 2004) were extended with word-level approaches (Ueffing and Ney, 2007) and more recently with gaussian processes by Beck et al. (2016). Mejer and Crammer (2011) proposed an uncertainty estimation model for structured predictions like sequence labelling and dependency parsing. Kochkina and Liakata (2020) modelled uncertainty in rumour verification models applying methods introduced by Kendall and Gal (2017) for computer vision. Xiao and Wang (2019) showed that accounting for uncertainty can lead to improved performance in sentiment analysis and named entity recognition, while Dong et al. (2018) focused on neural semantic parsing. Compared to the Bayesian setup, our approach has the obvious advantage of not requiring any complex assumption about prior distribution of the data, while at the same time being able to output valid predictions.

Recent studies combined conformal prediction with transformer architectures to create models having faster inference (Schuster et al., 2021) and better efficiency (Fisch et al., 2021).

3. Background

This section covers the concepts underpinning our experiments, starting with the theory of conformal prediction and some of its special cases. We will then briefly describe BERT as a Transformer-based model and how we combined these concepts together to build our system.

3.1. Conformal Prediction

Conformal prediction (CP) is a machine learning framework that was first introduced in Gammerman et al. (1998) and developed in a book by Vovk et al. (2005). Conformal predictors are guaranteed to be *valid*: given a significance level $\epsilon \in [0, 1]$, they make mistakes at a rate that is never higher than $(1 - \epsilon)$. The only assumption needed for the validity property to hold is for the data to be independent and identically distributed (IID) – more precisely, exchangeable. Conformal predictors are *set predictors*: they output a subset of labels Γ^ϵ if there is not enough information to output just a single label. The higher the requested confidence, the larger is the prediction set; obviously when we require 100% confidence it is very likely to end up with a trivial prediction that returns all the possible labels. A wrong prediction occurs when the true label of an example is not included in the prediction set.

The key mechanism of CP is to define a measure of “strangeness” (or *nonconformity measure* – NCM) for the training data points and use that NCM to assess how “strange” a new, unseen example (X, Y) is compared to the training ones. The *nonconformity score* $\alpha^{(y)}$ of the test example is calculated for each candidate label y. Then, in order to estimate how well the new test examples fit to the training set, the nonconformity scores are transformed into the statistical notion of p-values:

$$p_y = \frac{|\{i = 1, \dots, t : \alpha_i \geq \alpha_{t+1}^{(y)}\}| + 1}{t + 1} \tag{1}$$

where t is the number of training examples and α_{t+1} is the nonconformity score of the new test example. Once p-values are computed for each label, only those labels y for which $p_y \geq \epsilon$ are included in the output prediction set Γ^ϵ.

In some cases – especially when t is relatively small – the *smoothed* version of p-value may be preferred:

$$p_y^{\text{smooth}} = \frac{|\{i = 1, \ldots, t : \alpha_i > \alpha_{t+1}^{(y)}\}| + \tau_t|\{i = 1, \ldots, t : \alpha_i = \alpha_{t+1}^{(y)}\}| + 1}{t+1} \tag{2}$$

where τ_t is a random amount between 0 and $1/t$. This version of p-value is more careful in the management of the ties between nonconformity scores.

The computation of $\alpha^{(y)}$ can be based on any machine learning algorithm. For example, a nearest neighbours model can assess the strangeness of a new example depending on its distance from the nearest example of the same class. A support-vector machine can rate a new example stranger the higher is its Lagrangian multiplier (that is, the distance from the margin between classes).

This classical version of CP is known as transductive conformal prediction (TCP). Several modifications of CP have been introduced to address its limitations.

Inductive CP A limitation of TCP lays in its computational complexity: for every new test example, the underlying algorithm needs to be re-trained on the past examples. This is particularly inconvenient when using deep neural networks, as they can be very slow to train compared to other ML algorithms. To overcome this issue, Papadopoulos et al. (2002) introduced an *inductive* variant of conformal prediction (ICP)[1] where a *proper training set* is used only once to train the algorithm that will act as nonconformity measure; a smaller *calibration set* will instead be used to compute the nonconformity score of each new test example. ICP retains the validity property of standard CP while keeping the computational cost almost the same as the underlying algorithm alone.

Mondrian CP Standard CP does not guarantee validity within labels: in other words, a lower error rate for a label may compensate a higher error rate for another label in such a way that predictions are still valid overall. This can be an issue in those cases where the dataset is imbalanced or where wrong predictions are more impactful for one label over the others (*asymmetric classification*). Mondrian conformal predictors (Vovk et al., 2005) calculate nonconformity scores $\alpha^{(y)}$ against examples of the y label exclusively: this way validity is guaranteed for each label (*conditional validity*) so that, if needed, we may request a different significance ϵ for each label y.

Cross-conformal Prediction ICP is computationally efficient at the price of sacrificing part of the training set to build a calibration set. This may prove problematic when working with small datasets. Cross-conformal Prediction (XCP, Vovk, 2015) offers a workaround inspired by cross-validation: the training set is divided in K partitions and each of them is used in turn as calibration set, while the union of the remaining $K - 1$ partitions is used as proper training set. The K p-values obtained are then averaged together. While XCP cannot be proved to be valid in theory, empirical results show that its predictions are indeed valid and usually more efficient (tight) than those of standard CP. XCP predictors also are available in their Mondrian version.

[1] Otherwise known as *split* conformal prediction.

3.2. Transformer-based models

Introduced by Vaswani et al. (2017), transformers are neural architectures ideated for sequence modelling problems in NLP, such as machine translation. They are built around the idea of *self-attention*, a mechanism that allows to express each word in a sentence as a weighted combination of the other words in the same sentence (see Appendix A).

Transformers follow an encoder-decoder structure: the encoder's task is to learn a good representation for each word as a d-dimensional dense vector, while the decoder learns how to turn those representations into a new sequence of words[2] (see Appendix B for a more detailed introduction). Devlin et al. (2019) chose instead to use the sole encoder of a transformer and train it for two simple NLP tasks: guessing the missing word (i.e. a *cloze task*) and predicting whether or not a certain sentence b is likely to follow a sentence a. The training examples were automatically generated from a large, unlabelled text corpus (Wikipedia and BooksCorpus, see Zhu et al., 2015). Once trained, the model could be used as a starting point to build new models for specific NLP tasks (such as sequence classification) with minimal changes in the original architecture. The base model was named BERT (Bidirectional Encoder Representations from Transformers) and the two training steps would be known as pre-training and fine-tuning respectively.

BERT and its state-of-the-art performance on NLP tasks inspired the design of many variants that would focus on improving the model's training strategy (Liu et al., 2019), speed (Lan et al., 2020), size (Sanh et al., 2019), maximum example length (Beltagy et al., 2020) or even replacing the entire attention mechanism with simpler transformations (Lee-Thorp et al., 2021). Some architectures are based on transformers' decoders, such as OpenAI GPT-2 (Radford et al., 2019) or Reformer (Kitaev et al., 2020). Others, like BART (Lewis et al., 2020) are based on a combination of both encoder and decoder. The approach we propose can be applied to any of these models. In this work, we chose to use the original BERT model.

Much of BERT's versatility relies on the use of special tokens at the beginning of the encoding procedure. After tokenizing each example into a sequence of *wordpieces* (Schuster and Nakajima, 2012), each sentence in a pair is terminated by a `[SEP]` token; the concatenation of the two sentences is then padded with `[PAD]` tokens up to a fixed amount, while a further `[CLS]` token is inserted at the very start:

```
[CLS] my dog is cute [SEP] he likes play ##ing [SEP] [PAD]...[PAD]
```

Every sentence pair is thus treated as a single sequence. Each token is then transformed into a dense embedding $\boldsymbol{e} \in \mathbb{R}^h$ through a matrix which is learned at training time; the whole sequence is then represented as a matrix of embeddings and transformed layer by layer into a final representation where every token is of the form $\boldsymbol{w} \in \mathbb{R}^d$.

Even the `[CLS]` token will get its dense representation $\boldsymbol{w}_{\text{CLS}} \in \mathbb{R}^d$. This special token is fully connected to a K-neuron output layer where K is the number of labels: during the fine-tuning phase, $\boldsymbol{w}_{\text{CLS}}$ will be updated so that the log loss of the output layer is minimized. For this reason, $\boldsymbol{w}_{\text{CLS}}$ can be seen as a dense representation of the whole sequence.

[2] With the term "word" we denote any kind of symbol.

3.3. BERT as a nonconformity measure

A fine-tuned BERT model produces a *logit* $z^{(k)} \in \mathbb{R},\ k = 1, \dots, K$ for each of the K labels present in the dataset. Intuitively, the higher the logit, the higher is the probability of the related label to be the true label. However, logits do not add up to 1, so it is hard to quantify the probability of a given prediction to be correct. For this reason, logits are often passed to the softmax function:

$$\text{softmax}(z^{(k)}) = \frac{e^{z^{(k)}}}{\sum_{i=1}^{K} e^{z^{(i)}}} \qquad \text{for } k = 1, \dots, K. \tag{3}$$

While softmax scores of a prediction sum up to 1 and as such they resemble probability estimates, they are not guaranteed to reflect actual probabilities, where by "actual probability" we indicate the rate of correct predictions. In fact, Guo et al. (2017) and, specifically for NLP, Vasudevan et al. (2019) showed how modern deep neural architectures often suffer from poor calibration. Our approach is to set the desired error rate first and then use BERT outputs as nonconformity scores, regardless of them being logit scores or softmax scores.

A simple nonconformity score we can extract from BERT is the *negative logit*: if $\hat{z}_i$ is the logit computed by BERT for the true label of the i-th example,

$$\alpha_i = -\hat{z}_i \tag{4}$$

is a nonconformity score.

This choice differs from Maltoudoglou et al. (2020)'s nonconformity measure, where raw logits are passed to either a softmax or a sigmoid function; the resulting scores $\hat{\boldsymbol{y}}$ are subtracted from the true labels $\boldsymbol{y}$, then the nonconformity score is given by:

$$\alpha_i = \|\hat{\boldsymbol{y}}_i - \boldsymbol{y}_i\|_\infty \tag{5}$$

i.e. the absolute value of the largest difference between true score and predicted score in a single prediction.[3]

For simplicity, throughout this paper we will refer to the two nonconformity measures as logit and softmax respectively.

4. Experiments

We experiment with four conformal predictors, each based on both logit (4) and softmax (5) nonconformity measures : a standard inductive conformal predictor ICP, its Mondrian variant MICP, a cross-conformal predictor XCP and its Mondrian variant MXCP. For the cross-conformal setting we choose $K = 10$ folds.

4.1. Dataset

We run our experiments on the Microsoft Research Paraphrase Corpus (MRPC, Dolan and Brockett, 2005), a set of 5801 sentence pairs (a, b) extracted from news websites and labelled as whether they are semantically equivalent or not.[4] Each example $[(a, b), y]$ was actually

[3] Here the notion of L^∞ norm is used: $\|\boldsymbol{v}\|_\infty = \max\{|v_1|, |v_2|\}$.

[4] The authors define a paraphrase as a couple of sentences that exhibit "mostly bidirectional entailment".

generated by a trained SVM and then re-labelled by two human annotators, with a third judge to resolve conflicts. In terms of data distribution, we remark that:

- The label distribution in MRPC is imbalanced because the generating SVM was tuned to over-recognise positive examples
- As noted by Weeds et al. (2005), MRPC examples are very likely to present high overlap between sentence a and b (the two sentences share on average 70% of the words)

We do not know for sure if examples in MRPC are exchangeable – a property we need for CP to work as expected. In any case, studying the empirical validity of our predictors will provide more information about the data distribution.

We keep the original split of training set X_{train}, validation set X_{val} and test set X_{test} – respectively made of 3668, 408 and 1725 examples – as it was released by Microsoft. We also reserve 10% of X_{train} to be used as calibration set for the inductive conformal predictor.

4.2. Evaluation metrics

Since the user can set the required confidence level (or required error rate) in advance, it is not very useful to evaluate the performance of a conformal predictor in terms of accuracy or precision. Setting $\epsilon = 0.01$ is always going to result in a model that is wrong 1% of the time. However, what makes a difference between conformal predictors is how large their prediction regions Γ^{ϵ} are on average: a predictor that always outputs all the labels is not that useful, even if it is 0% wrong and valid. Hence we are going to evaluate our CP models by their *efficiency*, that is their ability to obtain small prediction regions. We use the following measures of efficiency described in Vovk et al. (2016).

The *S-criterion* is given by the average of the p-values of all the test examples:

$$S = \frac{1}{n} \sum_{i=1}^{n} \sum_{y} p_i^y \tag{6}$$

where n is the number of test examples and p^y is the p-value for label y. Smaller values are preferable, since ideally the majority of p-values should not be too large (see also Fedorova et al., 2013).

The *OF-criterion*, where OF stands for "observed fuzziness", also known as average false p-value, is defined as

$$OF = \frac{1}{n} \sum_{i=1}^{n} \sum_{y \neq y^{\star}} p_i^y \tag{7}$$

where n is the number of test examples, p^y is the p-value for label y and $y^{\star}$ is the true label of an example. Because the true label is included in Γ^{ϵ} with high probability, we would like the p-values of the false labels to be as small as possible. The smaller these false p-values, the tighter are the prediction regions on average, meaning that the predictor is more efficient.

Finally, the *N-criterion* measures the average sizes of the prediction region over all test examples:

$$N = \frac{1}{n} \sum_{i=1}^{n} |\Gamma_i^\epsilon| \tag{8}$$

where n is the number of test examples and Γ_i^ϵ is the prediction region for test example i at the significance level ϵ.

Although CP is designed to return prediction sets, sometimes it is useful to output a single label for each example. A way to do so in the CP framework is to choose the label with the highest p-value. The accuracy of these *forced predictions* can be evaluated with the standard methods for classification like precision, recall or F_1 score. Given the imbalance in our dataset labels we choose the Macro F_1 score to average the performance over both labels. Macro F_1 is defined as the arithmetic mean of the F_1 scores computed for each label. The F_1 score for a label k is defined as

$$F_1^{(k)} = \frac{2P^{(k)}R^{(k)}}{P^{(k)} + R^{(k)}} \tag{9}$$

where P and R are precision and recall scores.[5] More precisely, F_1 is the balanced version of the weighted harmonic mean of precision and recall (or balanced F measure):

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} \tag{10}$$

where $\alpha = 1/2$.

Macro F_1 assigns the same importance to each label regardless of how frequently it appears in the dataset – it is therefore recommended in the case of imbalanced datasets (see Manning et al., 2008, section 13.6, or Opitz and Burst, 2019). Conversely, other versions of F_1, such as Micro F_1 or "weighted" F_1 tend to be skewed towards the more represented label.

4.3. Training setup

We use the *base*, *uncased* pre-trained version of BERT. This version of BERT was trained on lower-cased English text and it is composed of 12 self-attention layers, for a total of 110 million parameters to estimate.

We fine-tune BERT on the MRPC training set in order to obtain our nonconformity measure. We minimize the log loss (or cross-entropy) using the AdamW optimizer (Loshchilov and Hutter, 2019) with a learning rate that follows a linear schedule with warm-up. We train for 2 epochs, so that even XCP – the slowest model – can be trained in a relatively short time. For each epoch we check the Macro F_1 score obtained on X_{val} and finally we select the model with the highest score.

Devlin et al. (2019) noted from the very start that fine-tuning BERT showed a highly variable performance on certain datasets. This applies to MRPC, where such behaviour persists even with fixed hyper-parameters, the random seed being the only variable (Dodge

[5] Given a set of predictions, if TP is the number of true positives, FP of false positives and FN of false negatives, precision is defined as $P = \frac{TP}{TP+FP}$ while recall is $R = \frac{TP}{TP+FN}$.

et al., 2020). According to Mosbach et al. (2021), fine-tuning instability may be due to optimization issues. In any case, in order to produce more meaningful results we repeat the fine-tuning step 5 times – each starting with a different random seed – and report average values of Macro F_1, S and OF scores. However, this will not apply to our cross-conformal predictors, since their p-values are already averaged over 10 folds.

5. Results

We evaluate our models in terms of validity, predictive accuracy and efficiency. From a qualitative perspective, we will also include a few examples of predictions from the test set and study how well they conform to the training set.

5.1. Validity

We start by empirically evaluating the validity of our predictions for all CP configurations. Figure 1 shows how the standard setting provides overall valid predictions but does not guarantee *conditional validity*, that is validity within labels. For example, when $\epsilon = 0.50$ the prediction error rate is indeed 0.50; however, this corresponds to an error rate of 0.94 for the negative label and 0.27 for the positive label. The higher error rate of the negative label is balanced by the lower error rate associated to the positive label, which is also the most frequent in our dataset. When $\epsilon \geq 0.55$ the negative label is never included in any prediction set, resulting in a 100% error rate for the negative label.

The Mondrian setting, designed to achieve conditional validity, overcomes this limitation. Figure 2(b) shows how the error rates of *each* label grow linearly with the significance.

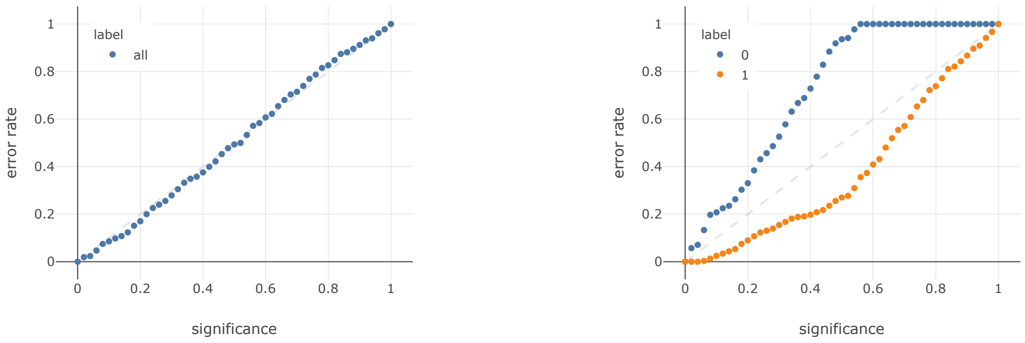
Unlike the two models just discussed, cross-conformal predictors are not theoretically guaranteed to be valid, so it is essential to check if at least their empirical validity holds for our dataset. Without validity, any consideration on the efficiency of a conformal predictor would be meaningless. In Appendix C we show that cross-conformal predictors produce figures similar to their non-cross counterparts, hence their validity is empirically verified.

5.2. Predictive accuracy

We fine-tuned BERT for only two epochs and without hyperparameter tuning, yet we obtained a fairly accurate model. The reported scores for BERT (12 layers) on GLUE[6] are 0.89/0.85 for F_1 score of the positive label and accuracy; our fine-tuned model scored a reasonable 0.87/0.82. Unfortunately GLUE does not report Macro F_1 scores which would be preferable (and likely lower) for an imbalanced dataset such as MRPC. In any case, we can say our fine-tuned BERT model is good enough for our task. In general, a better tuned BERT would probably result in better CP performances.

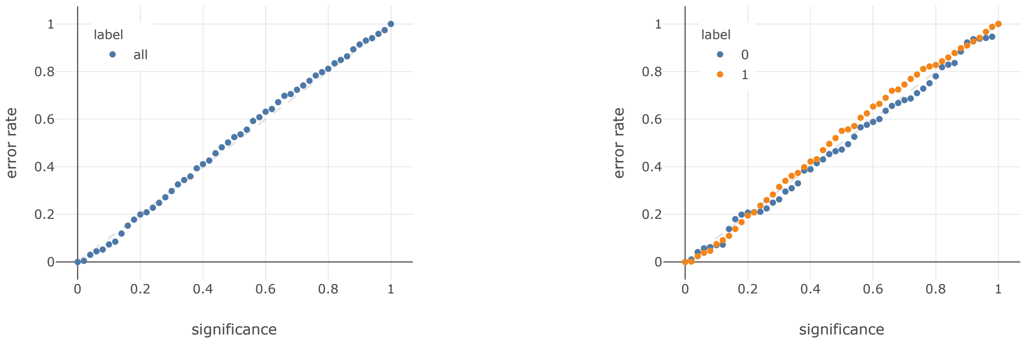
Macro F_1 scores for the forced predictions of the different models are shown in the first column of Table 1. We are not surprised to find the cross-conformal predictors as the high scorers (however not by a big margin), since XCP does not need to hold out a part of the training set for calibration purposes – each training example is seen by 9 of the 10 models. A downside of this approach is of course the computing time: our 10-fold XCP takes ten

[6] https://gluebenchmark.com/leaderboard



(*a*) (*b*)

Figure 1: Standard inductive conformal predictor: empirical validity check in the (*a*) unconditional and (*b*) label-conditional case on the MRPC test set. Labels 0 and 1 correspond to negative and positive labels respectively.



(*a*) (*b*)

Figure 2: Mondrian inductive conformal predictor: empirical validity check in the (*a*) unconditional and (*b*) label-conditional case on the MRPC test set.

		Macro F1	**S**	**OF**
BERT		0.773	-	-
logit	ICP	0.776	0.305	0.115
	MICP	0.771	0.317	0.135
	XCP	0.787	0.305	0.113
	MXCP	0.792	0.313	0.127
softmax	ICP	0.776	0.304	0.114
	MICP	0.772	0.317	0.135
	XCP	0.788	0.304	0.111
	MXCP	0.794	0.311	0.125

Table 1: Performance of different CP configurations on the MRPC test set. Macro F_1 measures the accuracy of forced predictions (a forced prediction is the label with the highest p-value). S and OF criteria measure the efficiency of a conformal predictor (lower scores are better). Values are averaged over 5 runs with different seeds. See Sections 4.2–4.3 for more details.

times more to train than its competitors. For very large datasets, this solution may not be viable.

It is interesting to note how all conformal predictors perform on par or even better than the original BERT model. This is also true for ICP and MCP, despite they were trained on 90% of the data, and applies to both logit and softmax nonconformity measures. Applying any kind of CP to the original Transformer model leaves predictive performance unharmed, while at the same time provides a valid measure of confidence.

5.3. Efficiency

The second and third columns of Table 1 show S-criterion and OF-criterion scores for the five different predictors. In accordance with the existing literature, Mondrian conformal predictors appear less efficient than their standard counterpart (smaller S- and OF-score). Cross-conformal predictors are the most efficient, even if not by a huge margin.

In terms of nonconformity scores, softmax achieves the best results overall. However, logit's performance is extremely close – the same, if we factor in statistical fluctuations – showing that raw BERT logits can act as a convenient nonconformity measure.

The N-criterion is dependent on the significance level ϵ, therefore it is appropriate to plot its value against different choices of ϵ. To mitigate BERT's instability (see Section 4.3) we compute the prediction region size as the average size over 5 runs. Figure 3 shows N-scores of the two inductive conformal predictors for each of the nonconformity measures (we restrict the plot to $\epsilon \leq 0.20$ for better visibility). Again, we observe that Mondrian predictors are typically the least efficient, while both logit and softmax show similar trends. The same plot for the cross-conformal setting is provided in Appendix C.

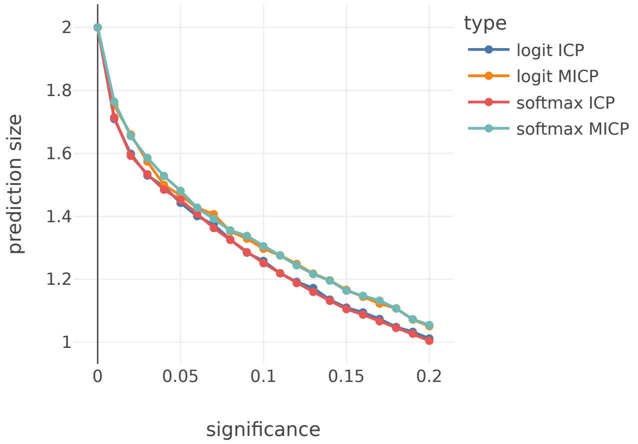


Figure 3: Average prediction set size (N-criterion) for the inductive conformal predictor (ICP), Mondrian ICP, cross-conformal Predictor (XCP) and Mondrian XCP.

5.4. Credibility

Credibility is defined as the largest p-value in a prediction set: an example is more conforming to the training set the higher is its credibility. A low-credibility example is an anomalous example for both labels – it may indicate the presence of a new, unobserved label or signal that the example is not IID. By analysing low-credibility predictions we may gain insight about the model's weaknesses or some property of the dataset. For each test example, its credibility will be the average credibility over 5 runs with different seeds.

Table 2 provides a few examples of low-credibility predictions taken from the MRPC test set. We speculate that non-paraphrases with many words in common are assigned low credibility. This sounds reasonable since – as we noted in Section 4.1 – high-overlap, positive-labelled sentence pairs make the majority of the dataset. To gain further insight, we analyse the relation between overlap and credibility.[7] Figure 7 in Appendix D shows that there is indeed a positive correlation between sentence overlap and credibility for positive examples. We cannot come to any conclusion about negative examples though: there is not such a correlation and credibility values are in general quite low. This is consistent with the low frequency of negative examples in the training set. The plots in Figure 8 show the same relationships in the Mondrian setting: in this case we see that there is indeed a negative correlation between overlap of negative examples and credibility.

[7] For two sentences a, b, if A, B are their word sets and $a \oplus b$ is their concatenation, their overlap is computed as $2 \times |A \cap B|/|a \oplus b|$.

Sentence A	**Sentence B**	**true**	**pred**	**cred**
Spokesmen for the FBI, CIA, Canadian Security Intelligence Service and Royal Canadian Mounted Police declined to comment on El Shukrijumah's stay in Canada.	The FBI, CIA, Canadian Security Intelligence Service and Royal Canadian Mounted Police declined to comment on the Washington Times report.	−	+	0.205
While waiting for a bomb squad to arrive, the bomb exploded, killing Wells.	The bomb exploded while authorities waited for a bomb squad to arrive.	−	+	0.207
Mrs. Clinton said she was incredulous that he would endanger their marriage and family.	She hadn't believed he would jeopardize their marriage and family.	+	−	0.213
The technology is available for download on the Microsoft Developer Network (MSDN) site.	WSE version 2 is available from Microsoft's developer Web site.	+	−	0.214

Table 2: Low-credibility examples (Sentence A, Sentence B) extracted from MRPC test set as scored by a BERT-based ICP. Credibility is defined as the largest p-value in a prediction region.

6. Discussion

We addressed the problem of obtaining reliable confidence estimates for the paraphrase detection task (PD). We showed that applying any of the variants of conformal prediction to a pre-trained transformer model is a successful approach: the original model's predictive accuracy is retained while the number of wrong predictions can be controlled (that is, we are able to build a *valid* predictor). Performances are evaluated on the Microsoft Research Paraphrase Corpus, a well known PD dataset with imbalanced labels – a context where Mondrian CP is recommended.

Our nonconformity estimator, a fine-tuned BERT model, performed well enough after 2 training epochs. The reference scores for BERT, as reported on the GLUE leaderboard, are not far away – still, these numbers are only indicative given how much they are influenced by the random parameter initialisation. We wish GLUE included Macro F_1 scores in addition to accuracy and F_1 score for the positive label. This would seem appropriate given the imbalanced nature of the dataset.

It is important to remember that CP predictors are valid under the assumption of data examples being IID (more exactly, exchangeable). While this condition cannot be guaranteed for our dataset (see Section 4.1), it is nonetheless reasonable to assume so and the empirical results confirm this theory.

An additional advantage of reliable confidence estimates is that they make a transformer model more explainable: by looking at the low credibility test examples, it may be possible to infer which features make a prediction hard or easy; it provides also a good way of spotting debatable labels assigned by the annotators – an issue frequently found in NLP datasets. We presented a few examples of low-credibility predictions and included scatter plots that shed light on the relationship between data features and credibility.

Our encouraging results suggest that our method can be suitable for other NLP tasks. Future directions include experimenting with larger PD datasets and with other classification / regression tasks such as natural language inference, semantic similarity scoring and sentiment scoring. In addition, it would be interesting to run further experiments using more recent and effective transformer-based models.

Acknowledgments

This research was partially supported by the Amazon AWS award “Conformal Martingales for Change-Point Detection”.

BERT is made available by the `transformers` library (Wolf et al., 2019); MRPC was downloaded and pre-processed with the Python library `datasets` (Lhoest et al., 2021).

Finally, we thank Jay Alammar, Miguel Romero and Francisco Ingham for their helpful blog posts *The Illustrated Transformer*[8] and *Dissecting BERT*.[9]

Appendix A. The attention mechanism

The term “attention” referred to a technique that helps modelling sequences made its first appearance in Bahdanau et al. (2015). The authors altered a *sequence-to-sequence* architecture (Sutskever et al., 2014) so that every step in the output sequence Y could be influenced by different parts of the input sequence X.

The original sequence to sequence model is made of two RNNs. The first one is called *encoder* and processes a sentence $X = (x_1, \dots, x_m)$ one word at a time. At time step j the encoder reads a word $x_j \in X$ and a hidden state h_{j-1} coming from the previous step. It then outputs a new hidden state

$$h_j = f(x_j, h_{j-1})$$

where f is a nonlinear function. After m steps the final hidden state h_m should contain information about the whole sentence: we will denote it as the “context vector” c.

The second RNN is the *decoder* and will try to predict the target sequence $Y = (y_1, \dots, y_n)$, where n may be different from m. At each time step i, the decoder outputs a word y_i whose probability is

$$p(y_i \mid \{y_1, \dots, y_{i-1}\}, c) = g(y_{i-1}, s_i, c)$$

where s_i is the hidden state of the decoder and g is a nonlinear function. A limitation of this approach is that the all the information coming from X has to be squashed into a single vector, c.

Bahdanau et al. (2015) addressed this issue by letting the decoder decide which of the hidden states h_j to consider when producing the next word. Formally, we have that for each output word y_i:

$$p(y_i \mid \{y_1, \dots, y_{i-1}\}, X) = g(y_{i-1}, s_i, c_i)$$

[8] https://jalammar.github.io/illustrated-transformer/

[9] https://medium.com/dissecting-bert/dissecting-bert-part-1-d3c3d495cdb3

where c_i is now a weighted sum of the encoder hidden states:

$$c_i = \sum_{j=1}^{m} \alpha_{ij} h_j$$

The *attention weights* α_{ij} are calculated by a feed-forward neural network that is trained jointly with the two RNNs to predict some form of affinity between the output state s_{i-1} and the input hidden state h_j.

Appendix B. The Transformer

Since Sutskever et al. (2014) introduced their "sequence to sequence" learning model, *encoder-decoder* architectures have dominated the field of sequence modelling in NLP. The objective of sequence modelling is to produce a sequence of symbols Y given an input sequence X. In machine translation, for example, X may be a phrase written in English while Y could be its translation in German. In the sequence to sequence model an RNN, called encoder, is tasked with compressing information about X, while a second RNN, the decoder, learns to generate Y from the output of the encoder. Bahdanau et al. (2015) extended this architecture with an *attention* mechanism that helps the decoder focus on different parts of X for each symbol of Y (see Appendix A).

In order to remove the need for RNNs and their computational burden, Vaswani et al. (2017) designed an encoder-decoder architecture that relies *exclusively* on attention. The core of their architecture, which they named *Transformer*, is the computation of three m-rows matrices Q, K and V that are learned at training time so that the probability of producing the correct sequence is maximized. The attention mechanism proposed by the author is to calculate

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right) \cdot V$$

where $\sqrt{d_k}$ is a scaling factor that helps with the gradient calculation. QK^T is a $m \times m$ matrix where each row can be seen as the similarity between one word and all the other words in the sentence. These similarity scores are passed to a row-wise softmax function and the resulting matrix will weight each of the m words present in V (V is a $m \times d$ matrix). Attention is thus a way to express any word in a sentence as a weighted sum of the other words in the same sentence.

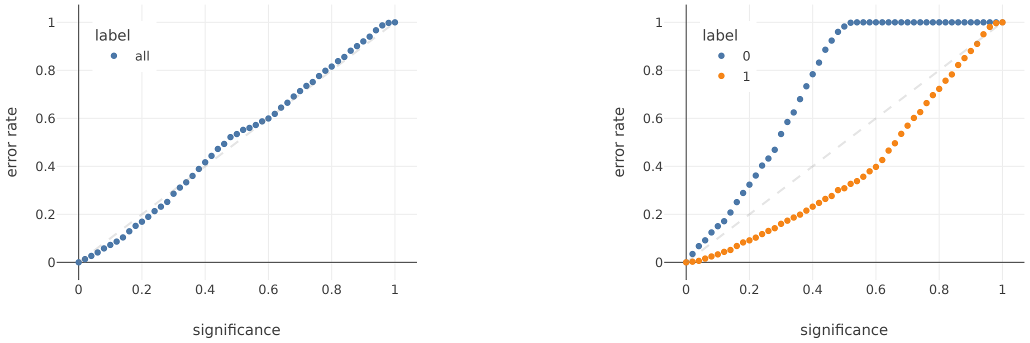
This kind of attention, where both input and output are the same sentence, is known as *self-attention*. The encoder is composed of several of these attention layers feeding one into the other.

The decoder is almost identical to the encoder. The main difference is that Q is now an $n \times d$ matrix, where each row corresponds to a word in the output sentence $Y = (y_1, \ldots, y_n)$. In this way it is possible to express each y_i as a weighted sum of all the input words x_i.

Transformers were shown to outperform state-of-the-art models on machine translation while being significantly faster to train, due to the approach based on matrix multiplication being easy to parallelize on GPU.

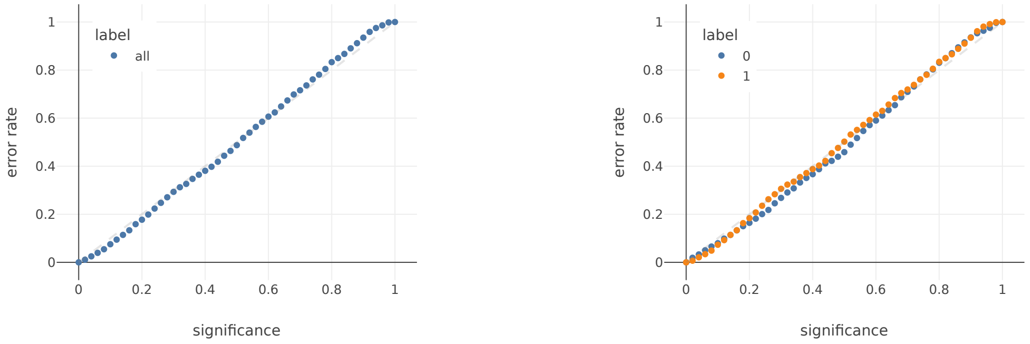
Appendix C. Validity end efficiency in the cross-conformal setting

We include plots for the empirical validity check of XCP (Figure 4) and MXCP (Figure 5). Figure 6 shows the N-criterion scores for the cross-conformal predictors.



(a) (b)

Figure 4: Cross-conformal predictor: empirical validity check in the (a) unconditional and (b) label-conditional case on the MRPC test set.



(a) (b)

Figure 5: Mondrian cross-conformal predictor: empirical validity check in the (a) unconditional and (b) label-conditional case on the MRPC test set.

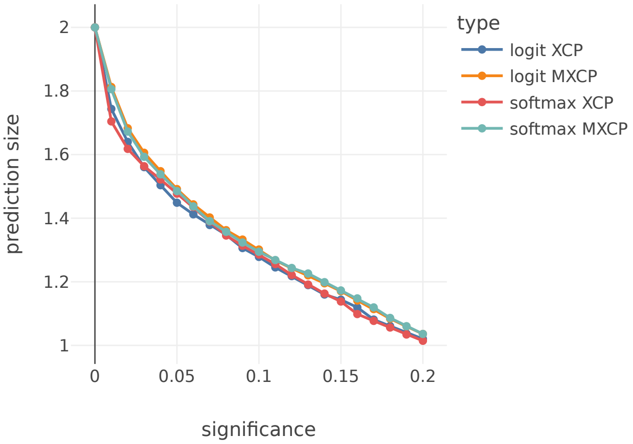


Figure 6: Average prediction set size (N-criterion) for the cross-conformal predictor (XCP) and its Mondrian version (MXCP) using two different nonconformity measures.

Appendix D. Credibility and overlap

The following scatter plots highlight the relation between overlap % and credibility of test examples (see Section 5.4). Figure 7 shows the standard setting while Figure 8 shows the Mondrian setting.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio and Yann LeCun, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. URL http://arxiv.org/abs/1409.0473.

Daniel Beck, Lucia Specia, and Trevor Cohn. Exploring prediction uncertainty in machine translation quality estimation. In *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, pages 208–218, Berlin, Germany, August 2016. Association for Computational Linguistics. doi: 10.18653/v1/K16-1021. URL https://www.aclweb.org/anthology/K16-1021.

Iz Beltagy, Matthew E Peters, and Arman Cohan. Longformer: The long-document transformer. *arXiv preprint arXiv:2004.05150*, 2020.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. Confidence estimation for machine translation.

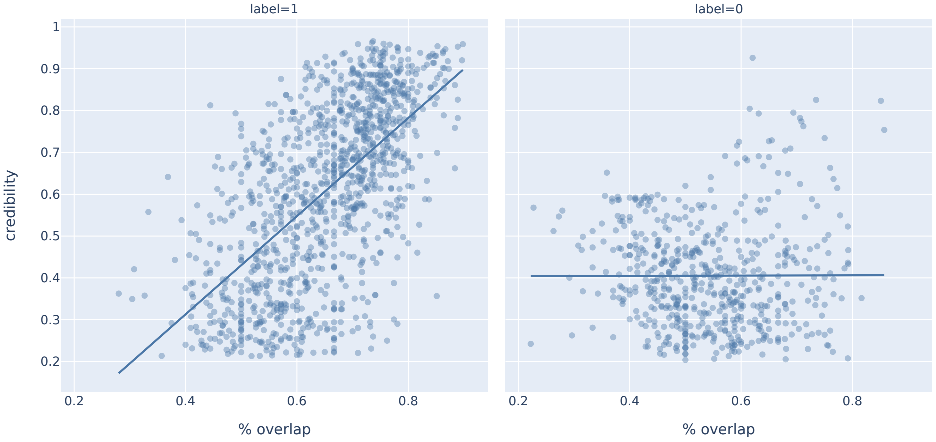


Figure 7: Effect of word overlap on credibility. Positive examples are expected to show high overlap. Negative examples, on the other hand, seem to have no relationship with word overlap.

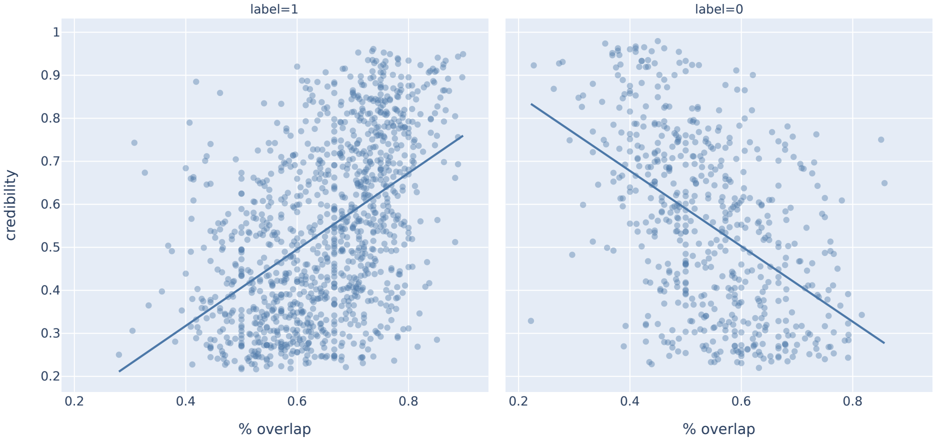


Figure 8: Effect of word overlap on credibility (Mondrian ICP). The Mondrian predictor is able to capture more of the distribution of negative examples: the less two sentences overlap, the more they are expected not to be paraphrases.

In *COLING 2004: Proceedings of the 20th International Conference on Computational Linguistics*, pages 315–321, Geneva, Switzerland, aug 23–aug 27 2004. COLING. URL https://www.aclweb.org/anthology/C04-1046.

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. URL https://www.aclweb.org/anthology/N19-1423.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, A. Farhadi, Hannaneh Hajishirzi, and Noah A. Smith. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *ArXiv*, abs/2002.06305, 2020.

William B. Dolan and Chris Brockett. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP2005)*, 2005. URL https://www.aclweb.org/anthology/I05-5002.

Li Dong, Chris Quirk, and Mirella Lapata. Confidence modeling for neural semantic parsing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 743–753, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1069. URL https://www.aclweb.org/anthology/P18-1069.

Tobias Falke, Markus Boese, Daniil Sorokin, Caglar Tirkaz, and Patrick Lehnen. Leveraging user paraphrasing behavior in dialog systems to automatically collect annotations for long-tail utterances. In *Proceedings of the 28th International Conference on Computational Linguistics: Industry Track*, pages 21–32, 2020.

Valentina Fedorova, Alex Gammerman, Ilia Nouretdinov, and Vladimir Vovk. Conformal prediction under hypergraphical models. In *IFIP International Conference on Artificial Intelligence Applications and Innovations*, pages 371–383. Springer, 2013.

Adam Fisch, Tal Schuster, Tommi S. Jaakkola, and Regina Barzilay. Efficient conformal prediction via cascaded inference with expanded admission. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=tnSo6VRLmT.

A. Gammerman, V. Vovk, and V. Vapnik. Learning by transduction. In *Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence*, UAI'98, page 148–155, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc. ISBN 155860555X.

Simona Gandrabur, George Foster, and Guy Lapalme. Confidence estimation for nlp applications. *ACM Trans. Speech Lang. Process.*, 3(3):1–29, October 2006. ISSN 1550-4875. doi: 10.1145/1177055.1177057. URL https://doi.org/10.1145/1177055.1177057.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine*

Learning, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330, International Convention Centre, Sydney, Australia, 06–11 Aug 2017. PMLR. URL http://proceedings.mlr.press/v70/guo17a.html.

Yun He, Zhuoer Wang, Yin Zhang, Ruihong Huang, and James Caverlee. PARADE: A New Dataset for Paraphrase Identification Requiring Computer Science Domain Knowledge. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7572–7582, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.611. URL https://www.aclweb.org/anthology/2020.emnlp-main.611.

Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5574–5584. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7141-what-uncertainties-do-we-need-in-bayesian-deep-learning-for-computer-vision.pdf.

Nikita Kitaev, Lukasz Kaiser, and Anselm Levskaya. Reformer: The efficient transformer. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkgNKkHtvB.

Elena Kochkina and Maria Liakata. Estimating predictive uncertainty for rumour verification models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6964–6981, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.623. URL https://www.aclweb.org/anthology/2020.acl-main.623.

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=H1eA7AEtvS.

James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontanon. Fnet: Mixing tokens with fourier transforms. *arXiv preprint arXiv:2105.03824*, 2021.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.703. URL https://www.aclweb.org/anthology/2020.acl-main.703.

Quentin Lhoest, Albert Villanova del Moral, Patrick von Platen, Thomas Wolf, Yacine Jernite, Abhishek Thakur, Lewis Tunstall, Suraj Patil, Mariama Drame, Julien Chaumond, Julien Plu, Joe Davison, Simon Brandeis, Teven Le Scao, Victor Sanh, Kevin Canwen

Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Steven Liu, Sylvain Lesage, Lysandre Debut, Théo Matussière, Clément Delangue, and Stas Bekman. huggingface/datasets: 1.11.0, July 2021. URL https://doi.org/10.5281/zenodo.5148649.

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. *arXiv preprint arXiv:1907.11692*, 2019.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=Bkg6RiCqY7.

Lysimachos Maltoudoglou, Andreas Paisios, and Harris Papadopoulos. Bert-based conformal predictor for sentiment analysis. volume 128 of *Proceedings of Machine Learning Research*, pages 269–284. PMLR, 09–11 Sep 2020. URL http://proceedings.mlr.press/v128/maltoudoglou20a.html.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to Information Retrieval*. Cambridge University Press, USA, 2008. ISBN 0521865719.

A. Mejer and K. Crammer. Confidence estimation in structured prediction. *ArXiv*, abs/1111.1386, 2011.

Marius Mosbach, Maksym Andriushchenko, and Dietrich Klakow. On the stability of fine-tuning {bert}: Misconceptions, explanations, and strong baselines. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=nzpLWnVAyah.

Juri Opitz and Sebastian Burst. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*, 2019.

Andreas Paisios, Ladislav Lenc, Jiří Martínek, Pavel Král, and Harris Papadopoulos. A deep neural network conformal predictor for multi-label text classification. volume 105 of *Proceedings of Machine Learning Research*, pages 228–245, Golden Sands, Bulgaria, 09–11 Sep 2019. PMLR. URL http://proceedings.mlr.press/v105/paisios19a.html.

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.

Alec Radford, Jeffrey Wu, R. Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

Mike Schuster and Kaisuke Nakajima. Japanese and korean voice search. In *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5149–5152, 2012. doi: 10.1109/ICASSP.2012.6289079.

Tal Schuster, Adam Fisch, Tommi Jaakkola, and Regina Barzilay. Consistent accelerated inference via confident adaptive transformers. *arXiv preprint arXiv:2104.08803*, 2021.

Artem Shelmanov, Evgenii Tsymbalov, Dmitri Puzyrev, Kirill Fedyanin, Alexander Panchenko, and Maxim Panov. How certain is your Transformer? In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1833–1840, Online, April 2021. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2021.eacl-main.157.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014. URL https://proceedings.neurips.cc/paper/2014/file/a14ac55a4f27472c5d894ec1c3c743d2-Paper.pdf.

Nicola Ueffing and Hermann Ney. Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics*, 33(1):9–40, 03 2007. ISSN 0891-2017. doi: 10.1162/coli.2007.33.1.9. URL https://doi.org/10.1162/coli.2007.33.1.9.

Vishal Thanvantri Vasudevan, Abhinav Sethy, and Alireza Roshan Ghias. Towards better confidence estimation for neural models. In *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7335–7339, 2019. doi: 10.1109/ICASSP.2019.8683359.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017. URL https://proceedings.neurips.cc/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

Vladimir Vovk. Cross-conformal predictors. *Annals of Mathematics and Artificial Intelligence*, 74(1):9–28, 2015.

Vladimir Vovk, Alex Gammerman, and Glenn Shafer. *Algorithmic learning in a random world*. Springer Science & Business Media, 2005. doi: https://doi.org/10.1007/b106715.

Vladimir Vovk, Valentina Fedorova, Ilia Nouretdinov, and Alexander Gammerman. Criteria of efficiency for conformal prediction. In *Symposium on conformal and probabilistic prediction with applications*, pages 23–39. Springer, 2016.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *International Conference on Learning Representations*, 2019. URL https://openreview.net/forum?id=rJ4km2R5t7.

Julie Weeds, David Weir, and Bill Keller. The distributional similarity of sub-parses. In *Proceedings of the ACL Workshop on Empirical Modeling of Semantic Equivalence and Entailment*, pages 7–12, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/W05-1202.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Yijun Xiao and William Yang Wang. Quantifying uncertainties in natural language processing tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01): 7322–7329, Jul. 2019. doi: 10.1609/aaai.v33i01.33017322. URL https://ojs.aaai.org/index.php/AAAI/article/view/4719.

Yinfei Yang, Yuan Zhang, Chris Tar, and Jason Baldridge. PAWS-X: A cross-lingual adversarial dataset for paraphrase identification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3687–3692, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1382. URL https://www.aclweb.org/anthology/D19-1382.

Yuan Zhang, Jason Baldridge, and Luheng He. PAWS: Paraphrase adversaries from word scrambling. 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 1298–1308, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1131. URL https://www.aclweb.org/anthology/N19-1131.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, page 19–27, USA, 2015. IEEE Computer Society. ISBN 9781467383912. doi: 10.1109/ICCV.2015.11. URL https://doi.org/10.1109/ICCV.2015.11.