Claim Check-Worthiness Detection as Positive Unlabelled Learning

Dustin Wright and **Isabelle Augenstein**
Dept. of Computer Science
University of Copenhagen
Denmark
{dw|augenstein}@di.ku.dk

Abstract

As the first step of automatic fact checking, claim check-worthiness detection is a critical component of fact checking systems. There are multiple lines of research which study this problem: check-worthiness ranking from political speeches and debates, rumour detection on Twitter, and citation needed detection from Wikipedia. To date, there has been no structured comparison of these various tasks to understand their relatedness, and no investigation into whether or not a unified approach to all of them is achievable. In this work, we illuminate a central challenge in claim check-worthiness detection underlying all of these tasks, being that they hinge upon detecting both how factual a sentence is, as well as how likely a sentence is to be believed without verification. As such, annotators only mark those instances they judge to be clear-cut check-worthy. Our best performing method is a unified approach which automatically corrects for this using a variant of positive unlabelled learning that finds instances which were incorrectly labelled as not check-worthy. In applying this, we outperform the state of the art in two of the three tasks studied for claim check-worthiness detection in English.

1 Introduction

Misinformation is being spread online at ever increasing rates (Del Vicario et al., 2016) and has been identified as one of society's most pressing issues by the World Economic Forum (Howell et al., 2013). In response, there has been a large increase in the number of organizations performing fact checking (Graves and Cherubini, 2016). However, the rate at which misinformation is introduced and spread vastly outpaces the ability of any organization to perform fact checking, so only the most salient claims are checked. This obviates the need for being able to automatically find check-worthy content online and verify it.

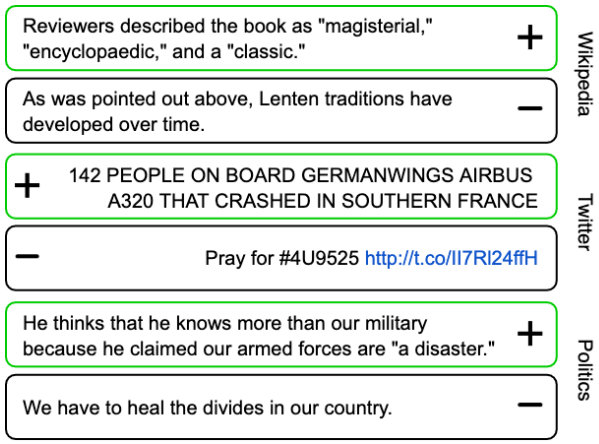

Figure 1: Examples of check-worthy and non check-worthy statements from three different domains. Check-worthy statements are those which were judged to require evidence or a fact check.

The natural language processing and machine learning communities have recently begun to address the problem of automatic fact checking (Vlachos and Riedel, 2014; Hassan et al., 2017; Thorne and Vlachos, 2018; Augenstein et al., 2019; Atanasova et al., 2020a,b; Ostrowski et al., 2020; Allein et al., 2020). The first step of automatic fact checking is claim check-worthiness detection, a text classification problem where, given a statement, one must predict if the content of that statement makes "an assertion about the world that is checkable" (Konstantinovskiy et al., 2018). There are multiple isolated lines of research which have studied variations of this problem. Figure 1 provides examples from three tasks which are studied in this work: rumour detection on Twitter (Zubiaga et al., 2016, 2018), check-worthiness ranking in political debates and speeches (Atanasova et al., 2018; Elsayed et al., 2019; Barrón-Cedeño et al., 2020), and citation needed detection on Wikipedia (Redi et al., 2019). Each task is concerned with a shared underlying problem: detecting claims which war-

rant further verification. However, no work has been done to compare all three tasks to understand shared challenges in order to derive shared solutions, which could enable improving claim check-worthiness detection systems across multiple domains.

Therefore, we ask the following main research question in this work: are these all variants of the same task, and if so, is it possible to have a unified approach to all of them? We answer this question by investigating the problem of annotator subjectivity, where annotator background and expertise causes their judgement of what is check-worthy to differ, leading to false negatives in the data (Konstantinovskiy et al., 2018). Our proposed solution is *Positive Unlabelled Conversion (PUC)*, an extension of Positive Unlabelled (PU) learning, which converts negative instances into positive ones based on the estimated prior probability of an example being positive. We demonstrate that a model trained using *PUC* improves performance on English *citation needed detection* and *Twitter rumour detection*. We also show that by pretraining a model on citation needed detection, one can further improve results on Twitter rumour detection over a model trained solely on rumours, highlighting that a unified approach to these problems is achievable. Additionally, we show that one attains better results on *political speeches* check-worthiness ranking without using any form of PU learning, arguing through a dataset analysis that the labels are much more subjective than the other two tasks.

The **contributions** of this work are as follows:

1. The first thorough comparison of multiple claim check-worthiness detection tasks.
2. *Positive Unlabelled Conversion (PUC)*, a novel extension of PU learning to support check-worthiness detection across domains.
3. Results demonstrating that a unified approach to check-worthiness detection is achievable for 2 out of 3 tasks, improving over the state-of-the-art for those tasks.

2 Related Work

2.1 Claim Check-Worthiness Detection

As the first step in automatic fact checking, claim check-worthiness detection is a binary classification problem which involves determining if a piece of text makes "an assertion about the world which can be checked" (Konstantinovskiy et al., 2018). We adopt this broad definition as it allows us to perform a structured comparison of many publicly available datasets. The wide applicability of the definition also allows us to study if and how a unified cross-domain approach could be developed.

Claim check-worthiness detection can be subdivided into three distinct domains: rumour detection on Twitter, check-worthiness ranking in political speeches and debates, and citation needed detection on Wikipedia. A few studies have been done which attempt to create full systems for mining check-worthy statements, including the works of Konstantinovskiy et al. (2018), ClaimRank (Jaradat et al., 2018), and ClaimBuster (Hassan et al., 2017). They develop full software systems consisting of relevant source material retrieval, check-worthiness classification, and dissemination to the public via end-user applications. These works are focused solely on the political domain, using data from political TV shows, speeches, and debates. In contrast, in this work we study the claim check-worthiness detection problem across three domains which have publicly available data: Twitter (Zubiaga et al., 2017), political speeches (Atanasova et al., 2018), and Wikipedia (Redi et al., 2019).

Rumour Detection on Twitter Rumour detection on Twitter is primarily studied using the PHEME dataset (Zubiaga et al., 2016), a set of tweets and associated threads from breaking news events which are either rumourous or not. Published systems which perform well on this task include contextual models (e.g. conditional random fields) acting on a tweet's thread (Zubiaga et al., 2017, 2018), identifying salient rumour-related words (Abulaish et al., 2019), and using a GAN to generate misinformation in order to improve a downstream discriminator (Ma et al., 2019).

Political Speeches For political speeches, the most studied datasets come from the Clef CheckThat! shared tasks (Atanasova et al., 2018; Elsayed et al., 2019; Barrón-Cedeño et al., 2020) and ClaimRank (Jaradat et al., 2018). The data consist of transcripts of political debates and speeches where each sentence has been annotated by an independent news or fact-checking organization for whether or not the statement should be checked for veracity. The most recent and best performing system on the data considered in this paper consists of a two-layer bidirectional GRU network which acts on both word embeddings and syntactic parse

tags (Hansen et al., 2019). In addition, they augment the native dataset with weak supervision from unlabelled political speeches.

Citation Needed Detection Wikipedia citation needed detection has been investigated recently in (Redi et al., 2019). The authors present a dataset of sentences from Wikipedia labelled for whether or not they have a citation attached to them. They also released a set of sentences which have been flagged as not having a citation but needing one (i.e. *unverified*). In contrast to other check-worthiness detection domains, there are much more training data available on Wikipedia. However, the rules for what requires a citation do not necessarily capture all "checkable" statements, as "all material in Wikipedia articles must be verifiable" (Redi et al., 2019). Given this, we view Wikipedia citation data as a set of positive and unlabelled data: statements which have attached citations are positive samples of check-worthy statements, and within the set of statements without citations there exist some positive samples (those needing a citation) and some negative samples. Based on this, this domain constitutes the most general formulation of check-worthiness among the domains we consider. Therefore, we experiment with using data from this domain as a source for transfer learning, training variants of PU learning models on it, then applying them to target data from other domains.

2.2 Positive Unlabelled Learning

PU learning methods attempt to learn good binary classifiers given only positive labelled and unlabelled data. Recent applications where PU learning has been shown to be beneficial include detecting deceptive reviews online (Li et al., 2014; Ren et al., 2014), keyphrase extraction (Sterckx et al., 2016) and named entity recognition (Peng et al., 2019). For a survey on PU learning, see (**?**), and for a formal definition of PU learning, see §3.2.

Methods for learning positive-negative (PN) classifiers from PU data have a long history (Denis, 1998; De Comité et al., 1999; Letouzey et al., 2000), with one of the most seminal papers being from Elkan and Noto (2008). In this work, the authors show that by assuming the labelled samples are a random subset of all positive samples, one can utilize a classifier trained on PU data in order to train a different classifier to predict if a sample is positive or negative. The process involves training a PN classifier with positive samples being shown to the classifier once and *unlabelled* samples shown as *both* a positive sample and a negative sample. The loss for the duplicated samples is weighted by the confidence of a PU classifier that the sample is positive.

Building on this, du Plessis et al. (2014) propose an unbiased estimator which improves the estimator introduced in (Elkan and Noto, 2008) by balancing the loss for positive and negative classes. The work of Kiryo et al. (2017) extends this method to improve the performance of deep networks on PU learning. Our work builds on the method of Elkan and Noto (2008) by relabelling samples which are highly confidently positive.

3 Methods

The task considered in this paper is to predict if a statement makes "an assertion about the world that is checkable" (Konstantinovskiy et al., 2018). As the subjectivity of annotations for existing data on claim check-worthiness detection is a known problem (Konstantinovskiy et al., 2018), we view the data as a set of positive and unlabelled (PU) data. In addition, we unify our approach to each of them by viewing Wikipedia data as an abundant source corpus. Models are then trained on this source corpus using variants of PU learning and transferred via fine-tuning to the other claim check-worthiness detection datasets, which are subsequently trained on as PU data. On top of vanilla PU learning, we introduce *Positive Unlabelled Conversion (PUC)* which relabels examples that are most confidently positive in the unlabelled data. A formal task definition, description of PU learning, and explanation of the *PUC* extension are given in the following sections.

3.1 Task Definition

The fundamental task is binary text classification. In the case of positive-negative (PN) data, we have a labelled dataset $\mathcal{D} : \{(x, y)\}$ with input features $x \in \mathbb{R}^d$ and labels $y \in \{0, 1\}$. The goal is to learn a classifier $g : x \to (0, 1)$ indicating the probability that the input belongs to the positive class. With PU data, the dataset $\mathcal{D}$ instead consists of samples $\{(x, s)\}$, where the value $s \in \{0, 1\}$ indicates if a sample is labelled or not. The primary difference from the PN case is that, unlike for the labels y, a value of $s = 0$ does not denote the sample is negative, but that the label is unknown. The goal is then to learn a PN classifier g using a PU classifier

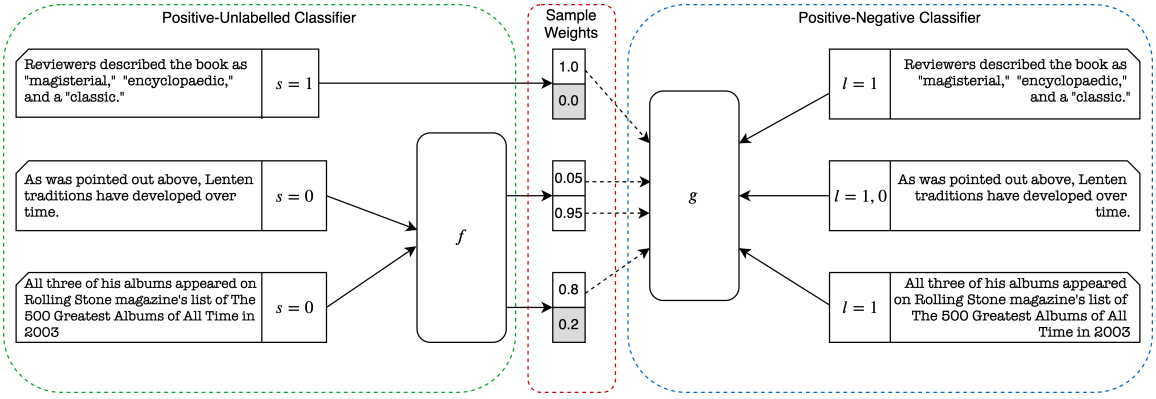

Figure 2: High level view of *PUC*. A PU classifier (f, green box) is first learned using PU data (with s indicating if the sample is positive or unlabelled). From this the prior probability of a sample being positive is estimated. Unlabelled samples are then ranked by f (red box) and the most positive samples are converted into positives until the dataset is balanced according to the estimated prior. The model g is then trained using the duplication and weighting method of Elkan and Noto (2008) as described in §3.2 with labels l (blue box). Greyed out boxes are negative weights which are ignored when training the classifier g, as those examples are only trained as positives.

$f : x \rightarrow (0, 1)$ which predicts whether or not a sample is labelled (Elkan and Noto, 2008).

3.2 PU Learning

Our overall approach is depicted in Figure 2. We begin with an explanation of the PU learning algorithm described in (Elkan and Noto, 2008). Assume that we have a dataset randomly drawn from some probability distribution $p(x, y, s)$, where samples are of the form (x, s), $s \in \{0, 1\}$ and $s = 1$ indicates that the sample is labelled. The variable y is unknown, but we make two assumptions which allow us to derive an estimator for probabilities involving y. The first is that:

$$p(y = 0|s = 1) = 0 \tag{1}$$

In other words, if we know that a sample is labelled, then that label cannot be 0. The second assumption is that labelled samples are Selected Completely At Random from the underlying distribution (also known as the SCAR assumption). Check-worthiness data can be seen as an instance of SCAR PU data; annotators tend to only label those instances which are very clearly check-worthy in *their* opinion (Konstantinovskiy et al., 2018). When combined across several annotators, we assume this leads to a random sample from the total set of check-worthy statements.

Given this, a classifier $f : x \rightarrow (0, 1)$ is trained to predict $p(s = 1|x)$ from the PU data. It is then employed to train a classifier g to predict $p(y = 1|x)$ by first estimating $c = p(s = 1|y = 1)$ on a set of validation data. Considering a validation set V where $P \subset V$ is the set of positive samples in V, c is estimated as:

$$c \approx \frac{1}{|P|} \sum_{x \in P} f(x) \tag{2}$$

This says our estimate of $p(s = 1|y = 1)$ is the average confidence of our classifier on known positive samples. Next, we can estimate $E_{p(x,y,s)}[h(x, y)]$ for any arbitrary function h empirically from a dataset of k samples as follows:

$$E[h] = \frac{1}{k}\Big(\sum_{(x,s=1)} h(x, 1) + \sum_{(x,s=0)} w(x)h(x, 1) + (1 - w(x))h(x, 0)\Big) \tag{3}$$

$$\begin{aligned} w(x) &= p(y = 1|x, s = 0) \\ &= \frac{1 - c}{c} \frac{p(s = 1|x)}{1 - p(s = 1|x)} \end{aligned} \tag{4}$$

In this case, c is estimated using Equation 2 and $p(s = 1|x)$ is estimated using the classifier f. The derivations for these equations can be found in (Elkan and Noto, 2008).

To estimate $p(y = 1|x)$ empirically, the unlabelled samples in the training data are duplicated, with one copy negatively labelled and one copy positively labelled. Each copy is trained on with a weighted loss $w(x)$ when the label is positive and $1 - w(x)$ when the label is negative. Labelled samples are trained on normally (i.e. a single copy with unit weight).

3.3 Positive Unlabelled Conversion

For *PUC*, the motivation is to relabel those samples from the unlabelled data which are very clear cut positive. To accomplish this, we start with the fact that one can also estimate the prior probability of a sample having a positive label using f. If instead of h we want to estimate $E[y] = p(y = 1)$, the following is obtained:

$$p(y = 1) \approx \frac{1}{k}\left(\sum_{x,s=1} 1 + \sum_{x,s=0} w(x)\right) \quad (5)$$

This estimate is then utilized to convert the most confident unlabelled samples into positives. First, all of the unlabelled samples are ranked according to their calculated weight $w(x)$. The ranked samples are then iterated through and converted into positive-only samples until the distribution of positive samples is greater than or equal to the estimate of $p(y = 1)$. Unlike in vanilla PU learning, these samples are discretized to have a positive weight of 1, and trained on by the classifier g once per epoch as positive samples along with the labelled samples. The remaining unlabelled data are trained on in the same way as in vanilla PU learning.

3.4 Implementation

In order to create a unified approach to check-worthiness detection, transfer learning from Wikipedia citation needed detection is employed. To accomplish this, we start with a training dataset $\mathcal{D}^s$ of statements from Wikipedia featured articles that are either labelled as containing a citation (positive) or unlabelled. We train a classifier f^s on this dataset and obtain a classifier g^s via *PUC*. For comparison, we also train models with vanilla PU learning and PN learning as baselines. The network architecture for both f^s and g^s is BERT (Devlin et al., 2019), a large pretrained transformer-based (Vaswani et al., 2017) language model. We use the HuggingFace transformers implementation of the 12-layer 768 dimensional variation of BERT (Wolf et al., 2019). The classifier in this implementation is a two layer neural network acting on the `[CLS]` token.

From g^s, we train a classifier g^t using downstream check-worthiness detection dataset D^t by initializing g^t with the base BERT network from g^s and using a new randomly initialized final layer. In addition, we train a model f^t on the target dataset, and train g^t with *PUC* from this model to obtain the final classifier. As a baseline, we also experiment with training on just the dataset D^t without any pretraining. In the case of citation needed detection, since the data comes from the same domain we simply test on the test split of statements labelled as "citation needed" using the classifier g^s. We compare our models to the published state of the art baselines on each dataset.

For all of our models (f^s, g^s, f^t, g^t) we train for two epochs, saving the weights with the best F1 score on validation data as the final model. Training is performed with a max learning rate of 3e-5 and a triangular learning rate schedule (Howard and Ruder, 2018) that linearly warms up for 200 training steps, then linearly decays to 0 for the rest of training. For regularization we add L2 loss with a coefficient of 0.01, and dropout with a rate of 0.1. Finally, we split the training sets into 80% train and 20% validation, and train with a batch size of 8. The code to reproduce our experiments can be found here.[1]

4 Experimental Results

To what degree is claim check-worthiness detection a PU learning problem, and does this enable a unified approach to check-worthiness detection? In our experiments, we progressively answer this question by answering the following: 1) is PU learning beneficial for the tasks considered? 2) Does PU citation needed detection transfer to rumour detection? 3) Does PU citation needed detection transfer to political speeches? To investigate how well the data in each domain reflects the definition of a check-worthy statement as one which "makes an assertion about the world which is checkable" and thus understand subjectivity in the annotations, we perform a dataset analysis comparing the provided labels of the top ranked check-worthy claims from the *PUC* model with the labels given by two human annotators. In all experiments, we report the mean performance of our models and standard deviation across 15 different random seeds. Additionally, we report the performance of each model ensembled across the 15 runs through majority vote on each sample.

4.1 Datasets[2]

Wikipedia Citations We use the dataset from Redi et al. (2019) for citation needed detection.

[1] https://github.com/copenlu/check-worthiness-pu-learning

[2] See supplemental material for links to datasets

Method	P	R	F1	eP	eR	eF1
Redi et al. 2019	75.3	70.9	73.0 [76.0]*	-	-	-
BERT	78.8 ± 1.3	83.7 ± 4.5	81.0 ± 1.5	79.0	85.3	82.0
BERT + PU	**78.8 ± 0.9**	84.3 ± 3.0	81.4 ± 1.0	79.0	85.6	82.2
BERT + *PUC*	78.4 ± 0.9	**85.6 ± 3.2**	**81.8 ± 1.0**	78.6	**87.1**	**82.6**

Table 1: F1 and ensembled F1 score for citation needed detection training on the FA split and testing on the LQN split of (Redi et al., 2019). The FA split contains statements with citations from featured articles and the LQN split consists of statements which were flagged as not having a citation but needing one. Listed are the mean, standard deviation, and ensembled results across 15 seeds (eP, eR, and eF1). **Bold** indicates best performance, underline indicates second best. *The reported value is from rerunning their released model on the test dataset. The value in brackets is the value reported in the original paper.

The dataset is split into three sets: one coming from featured articles (deemed 'high quality', 10k positive and 10k negative statments), one of statements which have no citation but have been flagged as needing one (10k positive, 10k negative), and one of statements from random articles which have citations (50k positive, 50k negative). In our experiments the models were trained on the high quality statements from featured articles and tested on the statements which were flagged as 'citation needed'. The key differentiating features of this dataset from the other two datasets are: 1) the domain of text is Wikipedia and 2) annotations are based on the decisions of Wikipedia editors following Wikipedia guidelines for citing sources[3].

Twitter Rumours The PHEME dataset of rumours is employed for Twitter claim checkworthiness detection (Zubiaga et al., 2016). The data consists of 5,802 annotated tweets from 5 different events, where each tweet is labelled as rumourous or non-rumourous (1,972 rumours, 3,830 non-rumours). We followed the leave-one-out evaluation scheme of (Zubiaga et al., 2017), namely, we performed a 5-fold cross-validation for all methods, training on 4 events and testing on 1. The key differentiating features of this dataset from the other two datasets are: 1) the domain of data is tweets and 2) annotations are collected from professional journalists specifically for building a dataset to train machine learning models.

Political Speeches The dataset we adopted in the political speeches domain is the same as in Hansen et al. (2019), consisting of 4 political speeches from the 2018 Clef CheckThat! competition (Atanasova et al., 2018) and 3 political speeches from ClaimRank (Jaradat et al., 2018) (2,602 statements total). We performed a 7-fold cross-validation, using 6 splits as training data and 1 as test in our experimental setup. The data from ClaimRank is annotated using the judgements from 9 fact checking organizations, and the data from Clef 2018 is annotated by factcheck.org. The key differentiating features of this dataset from the other two datasets are: 1) the domain of data is transcribed spoken utterances from political speeches and 2) annotations are taken from 9 fact checking organizations gathered independently.

[3] https://en.wikipedia.org/wiki/Wikipedia:Citing_sources

4.2 Is PU Learning Beneficial for Citation Needed Detection?

Our results for citation needed detection are given in Table 1. The vanilla BERT model already significantly outperforms the state of the art model from Redi et al. (2019) (a GRU network with global attention) by 6 F1 points. We see further gains in performance with PU learning, as well as when using *PUC*. Additionally, the models using PU learning have lower variance, indicating more consistent performance across runs. The best performing model we see is the one trained using *PUC* with an F1 score of 82.6. We find that this confirms our hypothesis that citation data is better seen as a set of positive and unlabelled data when used for checkworthiness detection. In addition, it gives some indication that PU learning improves the generalization power of the model, which could make it better suited for downstream tasks.

4.3 Does PU Citation Needed Detection Transfer to Rumour Detection?

4.3.1 Baselines

The best published method that we compare to is the CRF from (Zubiaga et al., 2017). which utilizes a combination of content and social features. Content features include word vectors, part-of-speech

Method	μP	μR	μF1	eP	eR	eF1
Zubiaga et al. 2017	66.7	55.6	60.7	-	-	-
BiLSTM	62.3	56.4	59.0	-	-	-
BERT	69.9 $\pm$ 1.7	60.8 $\pm$ 2.6	65.0 $\pm$ 1.3	71.3	61.9	66.3
BERT + Wiki	69.3 $\pm$ 1.6	61.4 $\pm$ 2.6	65.1 $\pm$ 1.2	70.7	62.2	66.2
BERT + WikiPU	69.9 $\pm$ 1.3	62.5 $\pm$ 1.6	66.0 $\pm$ 1.1	**72.2**	64.6	68.2
BERT + Wiki*PUC*	**70.1 $\pm$ 1.1**	61.8 $\pm$ 1.8	65.7 $\pm$ 1.0	71.5	62.7	66.8
BERT + PU	68.7 $\pm$ 1.2	64.7 $\pm$ 1.8	66.6 $\pm$ 0.9	69.9	65.2	67.5
BERT + *PUC*	68.1 $\pm$ 1.5	65.3 $\pm$ 1.6	66.6 $\pm$ 0.9	69.1	66.3	67.7
BERT + PU + WikiPU	68.4 $\pm$ 1.2	**66.1 $\pm$ 1.2**	**67.2 $\pm$ 0.6**	69.3	67.2	68.3
BERT + *PUC* + WikiPUC	68.0 $\pm$ 1.4	66.0 $\pm$ 2.0	67.0 $\pm$ 1.3	69.4	**67.5**	**68.5**

Table 2: micro-F1 (μF1) and ensembled F1 (eF1) performance of each system on the PHEME dataset. Performance is averaged across the five splits of (Zubiaga et al., 2017). Results show the mean, standard deviation, and ensembled score across 15 seeds. **Bold** indicates best performance, underline indicates second best.

tags, and various lexical features, and social features include tweet count, listed count, follow ratio, age, and whether or not a user is verified. The CRF acts on a timeline of tweets, making it contextual. In addition, we include results from a 2-layer BiLSTM with FastText embeddings (Bojanowski et al., 2017). There exist other deep learning models which have been developed for this task, including (Ma et al., 2019) and (Abulaish et al., 2019), but they do not publish results on the standard splits of the data and we were unable to recreate their results, and thus are omitted.

4.3.2 Results

The results for the tested systems are given in Table 2. Again we see large gains from BERT based models over the baseline from (Zubiaga et al., 2017) and the 2-layer BiLSTM. Compared to training solely on PHEME, fine tuning from basic citation needed detection sees little improvement (0.1 F1 points). However, fine tuning a model trained using PU learning leads to an increase of 1 F1 point over the non-PU learning model, indicating that PU learning enables the Wikipedia data to be useful for transferring to rumour detection i.e. the improvement is not only from a better semantic representation learned from Wikipedia data. For *PUC*, we see an improvement of 0.7 F1 points over the baseline and lower overall variance than vanilla PU learning, meaning that the results with *PUC* are more consistent across runs. The best performing models also use PU learning on in-domain data, with the best average performance being from the models trained using PU/*PUC* on in domain data and initialized with weights from a Wikipedia model trained using PU/*PUC*. When models are ensembled, pretraining with vanilla PU learning improves over no pretraining by almost 2 F1 points, and the best performing models which are also trained using PU learning on in domain data improve over the baseline by over 2 F1 points. We conclude that framing rumour detection on Twitter as a PU learning problem leads to improved performance.

Based on these results, we are able to confirm two of our hypotheses. The first is that Wikipedia citation needed detection and rumour detection on Twitter are indeed similar tasks, and a unified approach for both of them is possible. Pretraining a model on Wikipedia provides a clear downstream benefit when fine-tuning on Twitter data, *precisely when PU/PUC is used.* Additionally, training using *PUC* on in domain Twitter data provides further benefit. This shows that *PUC* constitutes a unified approach to these two tasks.

The second hypothesis we confirm is that both Twitter and Wikipedia data are better seen as positive and unlabelled for claim check-worthiness detection. When pretraining with the data as a traditional PN dataset there is no performance gain and in fact a performance loss when the models are ensembled. PU learning allows the model to learn better representations for general claim check-worthiness detection.

To explain why this method performs better, Table 1 and Table 2 show that *PUC* improves model recall at very little cost to precision. The aim of this is to mitigate the issue of subjectivity in the annotations of check-worthiness detection datasets noted in previous work (Konstantinovskiy et al., 2018). Some of the effects of this are illustrated in Table 5 and Table 6 in Appendix A The *PUC* models are better at distinguishing rumours which involve claims of fact about people i.e. things that people said or did, or qualities about people. For non-rumours, the *PUC* pretrained model is better at

Method	MAP
Konstantinovskiy et al. 2018	26.7
Hansen et al. 2019	30.2
BERT	33.0 ± 1.8
BERT + Wiki	**34.4 ± 2.7**
BERT + WikiPU	33.2 ± 1.7
BERT + Wiki*PUC*	31.7 ± 1.8
BERT + PU	18.8 ± 3.7
BERT + *PUC*	26.7 ± 2.8
BERT + PU + WikiPU	16.8 ± 3.5
BERT + *PUC* + Wiki*PUC*	27.8 ± 2.7

Table 3: Mean average precision (MAP) of models on political speeches. **Bold** indicates best performance, underline indicates second best.

Dataset	P	R	F1
Wikipedia	81.7	87.0	84.3
	84.8	87.0	85.9
	83.3	*87.0*	*85.1*
Twitter	87.5	82.4	84.8
	86.3	81.2	83.6
	86.9	*81.8*	*84.2*
Politics	33.8	89.3	49.0
	31.1	100.0	47.5
	32.5	*94.7*	*48.3*

Table 4: F1 score comparing manual relabelling of the top 100 predictions by *PUC* model with the original labels in each dataset by two different annotators. *Italics* are average value between the two annotators.

recognizing statements which describe qualitative information surrounding the events and information that is self-evident e.g. a tweet showing the map where the Charlie Hebdo attack took place.

4.4 Does PU Citation Needed Detection Transfer to Political Speeches?

4.4.1 Baselines

The baselines we compare to are the state of the art models from Hansen et al. (2019) and Konstantinovskiy et al. (2018). The model from Konstantinovskiy et al. (2018) consists of InferSent embeddings (Conneau et al., 2017) concatenated with POS tag and NER features passed through a logistic regression classifier. The model from Hansen et al. (2019) is a bidirectional GRU network acting on syntatic parse features concatenated with word embeddings as the input representation.

4.4.2 Results

The results for political speech check-worthiness detection are given in Table 3. We find that the BERT model initialized with weights from a model trained on plain Wikipedia citation needed statements performs the best of all models. As we add transfer learning and PU learning, the performance steadily drops. We perform a dataset analysis to gain some insight into this effect in §4.5.

4.5 Dataset Analysis

In order to understand our results in the context of the selected datasets, we perform an analysis to learn to what extent the positive samples in each dataset reflect the definition of a check-worthy claim as "an assertion about the world that is checkable". We ranked all of the statements based on the predictions of 15 *PUC* models trained with different seeds, where more positive class predictions means a higher rank (thus more check-worthy), and had two experts manually relabel the top 100 statements. The experts were informed to label the statements based on the definition of check-worthy given above. We then compared the manual annotation to the original labels using F1 score. Higher F1 score indicates the dataset better reflects the definition of check-worthy we adopt in this work. Our results are given in Table 4.

We find that the Wikipedia and Twitter datasets contain labels which are more general, evidenced by similar high F1 scores from both annotators ($>$ 80.0). For political speeches, we observe that the human annotators both found many more examples to be check-worthy than were labelled in the dataset. This is evidenced by examples such as *It's why our unemployment rate is the lowest it's been in so many decades* being labelled as not check-worthy and *New unemployment claims are near the lowest we've seen in almost half a century* being labelled as check-worthy in the same document in the dataset's original annotations. This characteristic has been noted for political debates data previously (Konstantinovskiy et al., 2018), which was also collected using the judgements of independent fact checking organizations (Gencheva et al., 2017). Labels for this dataset were collected from various news outlets and fact checking organizations, which may only be interested in certain types of claims such as those most likely to be false. This makes it difficult to train supervised machine learning models for general check-worthiness detection based solely on text content and document context due to labelling inconsistencies.

5 Discussion and Conclusion

In this work, we approached claim check-worthiness detection by examining how to unify three distinct lines of work. We found that check-worthiness detection is challenging in any domain as there exist stark differences in how annotators judge what is check-worthy. We showed that one can correct for this and improve check-worthiness detection across multiple domains by using positive unlabelled learning. Our method enabled us to perform a structured comparison of datasets in different domains, developing a unified approach which outperforms state of the art in 2 of 3 domains and illuminating to what extent these datasets reflect a general definition of check-worthy.

Future work could explore different neural base architectures. Further, it could potentially benefit all tasks to consider the greater context in which statements are made. We would also like to acknowledge again that all experiments have only focused on English language datasets; developing models for other, especially low-resource languages, would likely result in additional challenges. We hope that this work will inspire future research on check-worthiness detection, which we see as an under-studied problem, with a focus on developing resources and models across many domains such as Twitter, news media, and spoken rhetoric.

Acknowledgements


This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 801199.

References

Muhammad Abulaish, Nikita Kumari, Mohd Fazil, and Basanta Singh. 2019. A Graph-Theoretic Embedding-Based Approach for Rumor Detection in Twitter. In *IEEE/WIC/ACM International Conference on Web Intelligence*, pages 466–470.

Liesbeth Allein, Isabelle Augenstein, and Marie-Francine Moens. 2020. Time-Aware Evidence Ranking for Fact-Checking. *arXiv preprint arXiv:2009.06402*.

Pepa Atanasova, Alberto Barron-Cedeno, Tamer Elsayed, Reem Suwaileh, Wajdi Zaghouani, Spas Kyuchukov, Giovanni Da San Martino, and Preslav Nakov. 2018. Overview of the CLEF-2018 CheckThat! Lab on Automatic Identification and Verification of Political Claims. Task 1: Check-Worthiness. *arXiv preprint arXiv:1808.05542*.

Pepa Atanasova, Jakob Grue Simonsen, Christina Lioma, and Isabelle Augenstein. 2020a. Generating Fact Checking Explanations. In *ACL*, pages 7352–7364. Association for Computational Linguistics.

Pepa Atanasova, Dustin Wright, and Isabelle Augenstein. 2020b. Generating Label Cohesive and Well-Formed Adversarial Claims. In *Proceedings of EMNLP*. Association for Computational Linguistics.

Isabelle Augenstein, Christina Lioma, Dongsheng Wang, Lucas Chaves Lima, Casper Hansen, Christian Hansen, and Jakob Grue Simonsen. 2019. MultiFC: A Real-World Multi-Domain Dataset for Evidence-Based Fact Checking of Claims. In *EMNLP/IJCNLP (1)*, pages 4684–4696. Association for Computational Linguistics.

Alberto Barrón-Cedeño, Tamer Elsayed, Preslav Nakov, Giovanni Da San Martino, Maram Hasanain, Reem Suwaileh, and Fatima Haouari. 2020. CheckThat! at CLEF 2020: Enabling the Automatic Identification and Verification of Claims in Social Media. In *European Conference on Information Retrieval*, pages 499–507. Springer.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. In *EMNLP 2017*, pages 670–680.

Francesco De Comité, François Denis, Rémi Gilleron, and Fabien Letouzey. 1999. Positive and Unlabeled Examples Help Learning. In *International Conference on Algorithmic Learning Theory*, pages 219–230. Springer.

Michela Del Vicario, Alessandro Bessi, Fabiana Zollo, Fabio Petroni, Antonio Scala, Guido Caldarelli, H Eugene Stanley, and Walter Quattrociocchi. 2016. The Spreading of Misinformation Online. *Proceedings of the National Academy of Sciences*, 113(3):554–559.

François Denis. 1998. PAC Learning From Positive Statistical Queries. In *International Conference on Algorithmic Learning Theory*, pages 112–126. Springer.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-Training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT 2019*, pages 4171–4186.

Marthinus C Du Plessis, Gang Niu, and Masashi Sugiyama. 2014. Analysis of Learning From Positive and Unlabeled Data. In *Advances in Neural Information Processing Systems*, pages 703–711.

Charles Elkan and Keith Noto. 2008. Learning Classifiers From Only Positive and Unlabeled Data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Minin*, pages 213–220.

Tamer Elsayed, Preslav Nakov, Alberto Barrón-Cedeño, Maram Hasanain, Reem Suwaileh, Giovanni Da San Martino, and Pepa Atanasova. 2019. Overview of the CLEF-2019 CheckThat! Lab: Automatic Identification and Verification of Claims. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 301–321. Springer.

Pepa Gencheva, Preslav Nakov, Lluís Màrquez, Alberto Barrón-Cedeño, and Ivan Koychev. 2017. A Context-Aware Approach for Detecting Worth-Checking Claims in Political Debates. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*, pages 267–276, Varna, Bulgaria. INCOMA Ltd.

Lucas Graves and Federica Cherubini. 2016. The Rise of Fact-Checking Sites in Europe. *Reuters Institute for the Study of Journalism*.

Casper Hansen, Christian Hansen, Stephen Alstrup, Jakob Grue Simonsen, and Christina Lioma. 2019. Neural Check-Worthiness Ranking With Weak Supervision: Finding Sentences for Fact-Checking. In *Companion Proceedings of the 2019 World Wide Web Conference*, pages 994–1000.

Naeemul Hassan, Gensheng Zhang, Fatma Arslan, Josue Caraballo, Damian Jimenez, Siddhant Gawsane, Shohedul Hasan, Minumol Joseph, Aaditya Kulkarni, Anil Kumar Nayak, et al. 2017. ClaimBuster: the First-Ever End-to-End Fact-Checking System. *Proceedings of the VLDB Endowment*, 10(12):1945–1948.

Jeremy Howard and Sebastian Ruder. 2018. Universal Language Model Fine-tuning for Text Classification. pages 328–339.

Lee Howell et al. 2013. Digital Wildfires in a Hyperconnected World. *WEF report*, 3:15–94.

Israa Jaradat, Pepa Gencheva, Alberto Barrón-Cedeño, Lluís Màrquez, and Preslav Nakov. 2018. Claimrank: Detecting Check-Worthy Claims in Arabic and English. pages 26–30.

Ryuichi Kiryo, Gang Niu, Marthinus C du Plessis, and Masashi Sugiyama. 2017. Positive-Unlabeled Learning With Non-Negative Risk Estimator. In *Advances in Neural Information Processing Systems*, pages 1675–1685.

Lev Konstantinovskiy, Oliver Price, Mevan Babakar, and Arkaitz Zubiaga. 2018. Towards Automated Factchecking: Developing an Annotation Schema and Benchmark For Consistent Automated Claim Detection. *arXiv preprint arXiv:1809.08193*.

Fabien Letouzey, François Denis, and Rémi Gilleron. 2000. Learning From Positive and Unlabeled Examples. In *International Conference on Algorithmic Learning Theory*, pages 71–85. Springer.

Huayi Li, Zhiyuan Chen, Bing Liu, Xiaokai Wei, and Jidong Shao. 2014. Spotting Fake Reviews Via Collective Positive-Unlabeled Learning. In *2014 IEEE International Conference on Data Mining*, pages 899–904. IEEE.

Jing Ma, Wei Gao, and Kam-Fai Wong. 2019. Detect Rumors on Twitter by Promoting Information Campaigns With Generative Adversarial Learning. In *The World Wide Web Conference*, pages 3049–3055.

Wojciech Ostrowski, Arnav Arora, Pepa Atanasova, and Isabelle Augenstein. 2020. Multi-Hop Fact Checking of Political Claims. *arXiv preprint arXiv:2009.06401*.

Minlong Peng, Xiaoyu Xing, Qi Zhang, Jinlan Fu, and Xuan-Jing Huang. 2019. Distantly Supervised Named Entity Recognition using Positive-Unlabeled Learning. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2409–2419.

Miriam Redi, Besnik Fetahu, Jonathan Morgan, and Dario Taraborelli. 2019. Citation Needed: A Taxonomy and Algorithmic Assessment of Wikipedia's Verifiability. In *The World Wide Web Conference*, pages 1567–1578.

Yafeng Ren, Donghong Ji, and Hongbin Zhang. 2014. Positive Unlabeled Learning for Deceptive Reviews Detection. In *EMNLP 2014*, pages 488–498.

Lucas Sterckx, Thomas Demeester, Chris Develder, and Cornelia Caragea. 2016. Supervised Keyphrase Extraction as Positive Unlabeled Learning. In *EMNLP 2016*, pages 1–6.

James Thorne and Andreas Vlachos. 2018. Automated Fact Checking: Task Formulations, Methods and Future Directions. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3346–3359, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is All You Need. In *Advances in Neural Information Processing Systems*, pages 5998–6008.

Andreas Vlachos and Sebastian Riedel. 2014. Fact Checking: Task Definition and Dataset Construction. In *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science*, pages 18–22.

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

Arkaitz Zubiaga, Elena Kochkina, Maria Liakata, Rob Procter, Michal Lukasik, Kalina Bontcheva, Trevor Cohn, and Isabelle Augenstein. 2018. Discourse-Aware Rumour Stance Classification in Social Media Using Sequential Classifiers. *Information Processing & Management*, 54(2):273–290.

Arkaitz Zubiaga, Maria Liakata, and Rob Procter. 2017. Exploiting Context for Rumour Detection in Social Media. In *International Conference on Social Informatics*, pages 109–123. Springer.

Arkaitz Zubiaga, Maria Liakata, Rob Procter, Geraldine Wong Sak Hoi, and Peter Tolmie. 2016. Analysing How People Orient to and Spread Rumours in Social Media by Looking at Conversational Threads. *PloS one*, 11(3).

A Examples of PUC Improvements for Rumour Detection

Examples of improvements for rumour detection using *PUC* can be found in Table 5.

B Reproducibility

B.1 Computing Infrastructure

All experiments were run on a shared cluster. Requested jobs consisted of 16GB of RAM and 4 Intel Xeon Silver 4110 CPUs. We used a single NVIDIA Titan X GPU with 12GB of RAM.

B.2 Average Runtimes

See Table 7 for model runtimes.

B.3 Number of Parameters per Model

We used BERT with a classifier on top for each model which consists of 109,483,778 parameters.

B.4 Validation Performance

Validation performances for the tested models are given in Table 8.

B.5 Evaluation Metrics

The primary evaluation metric used was F1 score. We used the sklearn implementation of `precision_recall_fscore_support`, which can be found here: `https://scikit-learn.org/stable/modules/generated/sklearn.metrics.precision_recall_fscore_support.html`. Briefly:

$$p = \frac{tp}{tp + fp}$$

$$r = \frac{tp}{tp + fn}$$

$$F1 = \frac{2 * p * r}{p + r}$$

where tp are true positives, fp are false positives, and fn are false negatives.

Additionally, we used the mean average precision calculation from the Clef19 Check That! challenge for political speech data, which can be found here: `https://github.com/apepa/clef2019-factchecking-task1/tree/master/scorer` Briefly:

$$\text{AP} = \frac{1}{|P|} \sum_i \frac{tp(i)}{i}$$

Rumour text	nPUC	nBaseline
Germanwings co-pilot had serious depressive episode: Bild newspaper http://t.co/RgSTrehD21	13	5
Now hearing 148 passengers + crew on board the #A320 that has crashed in southern French Alps. #GermanWings flight. @BBCWorld	10	2
It appears that #Ferguson PD are trying to assassinate Mike Brown's character after literally assassinating Mike Brown.	13	5
#Ferguson cops beat innocent man then charged him for bleeding on them: http://t.co/u1ot9Eh5Cq via @MichaelDalynyc http://t.co/AGJW2Pid1r	9	2

Table 5: Examples of rumours which the *PUC* model judges correctly vs the baseline model with no pretraining on citation needed detection. n* is the number of models among the 15 seeds which predicted the correct label (rumour).

Non-Rumour text	nPUC	nBaseline
A female hostage stands by the front entrance of the cafe as she turns the lights off in Sydney. #sydneysiege http://t.co/qNfCMv9yZt	11	5
Map shows where gun attack on satirical magazine #CharlieHebdo took place in central Paris http://t.co/5AZAKumpNd http://t.co/ECFYztMVk9	10	4
"Hands up! Don't shoot!" #ferguson https://t.co/svCE1S0Zek	12	7
Australian PM Abbott: Motivation of perpetrator in Sydney hostage situation is not yet known - @9NewsAUS http://t.co/SI01B997xf	10	6

Table 6: Examples of non-rumours which the *PUC* model judges correctly vs the baseline model with no pretraining on citation needed detection. n* is the number of models among the 15 seeds which predicted the correct label (non-rumour).

$$\text{mAP} = \frac{1}{|Q|} \sum_{q \in Q} \text{AP}(q)$$

where P are the set of positive instances, $tp(i)$ is an indicator function which equals one when the ith ranked sample is a true positive, and Q is the set of queries. In this work Q consists of the ranking of statements from each split of the political speech data.

B.6 Links to Data

- Citation Needed Detection (Redi et al., 2019): https://drive.google.com/drive/folders/1zG6orf0_h2jYBvGvso1pSy3ikbNiW0xJ
- PHEME (Zubiaga et al., 2016): https://figshare.com/articles/PHEME_dataset_for_Rumour_Detection_and_Veracity_Classification/6392078.
- Political Speeches: We use the same 7 splits as used in (Hansen et al., 2019). The first 5 can be found here: http://alt.qcri.org/clef2018-factcheck/data/uploads/clef18_fact_checking_lab_submissions_and_scores_and_combinations.zip. The files can be found under "task1_test_set/English/task1-en-file(3,4,5,6,7)". The last two files can be found here: https://github.com/apepa/claim-rank/tree/master/data/transcripts_all_sources. The files are "clinton_acceptance_speech_ann.tsv" and "trump_inauguration _ann.tsv".

B.7 Hyperparameters

We found that good defaults worked well, and thus did not perform hyperparameter search. The hyperparameters we used are given in Table 9.

Method	Wikipedia	PHEME	Political Speeches
BERT	34m30s	14m25s	8m11s
BERT + PU	40m7s	20m40s	15m38s
BERT + *PUC*	40m8s	21m20s	15m32s
BERT + Wiki	-	14m28s	8m50s
BERT + WikiPU	-	14m25s	8m41s
BERT + Wiki*PUC*	-	14m28s	8m38s
BERT + PU + WikiPU	-	20m41s	15m32s
BERT + *PUC* + WikiPUC	-	21m52s	15m40s

Table 7: Average runtime of each tested system for each split of the data

Method	Wikipedia	PHEME	Political Speeches
BERT	88.9	81.6	31.3
BERT + PU	89.0	83.7	18.2
BERT + *PUC*	89.2	82.8	32.0
BERT + Wiki	-	80.8	32.3
BERT + WikiPU	-	82.0	35.7
BERT + Wiki*PUC*	-	80.4	34.3
BERT + PU + WikiPU	-	82.9	33.3
BERT + *PUC* + WikiPUC	-	84.1	34.0

Table 8: Validation F1 performances for each tested model.

Hyperparameter	Value
Learning Rate	3e-5
Weight Decay	0.01
Batch Size	8
Dropout	0.1
Warmup Steps	200
Epochs	2

Table 9: Validation F1 performances used for each tested model.