


ARTICLE

Towards universal methods for fake news detection

Maria Pszona¹, Maria Janicka¹, Grzegorz Wojdyga² and Aleksander Wawer^{1,2,*} 

¹Samsung R&D Institute Poland, pl. Europejski 1, 00-844 Warsaw, Poland and ²Institute of Computer Science Polish Academy of Sciences, ul. Jana Kazimierza 5, 01-248 Warszawa, Poland

*Corresponding author. E-mail: axw@ipipan.waw.pl

(Received 24 February 2020; revised 10 September 2022; accepted 12 September 2022;
first published online 26 October 2022)

Abstract

Fake news detection is an emerging topic that has attracted a lot of attention among researchers and in the industry. This paper focuses on fake news detection as a text classification problem: on the basis of five publicly available corpora with documents labeled as true or fake, the task was to automatically distinguish both classes without relying on fact-checking. The aim of our research was to test the feasibility of a universal model: one that produces satisfactory results on all data sets tested in our article. We attempted to do so by training a set of classification models on one collection and testing them on another. As it turned out, this resulted in a sharp performance degradation. Therefore, this paper focuses on finding the most effective approach to utilizing information in a transferable manner. We examined a variety of methods: feature selection, machine learning approaches to data set shift (instance re-weighting and projection-based), and deep learning approaches based on domain transfer. These methods were applied to various feature spaces: linguistic and psycholinguistic, embeddings obtained from the Universal Sentence Encoder, and GloVe embeddings. A detailed analysis showed that some combinations of these methods and selected feature spaces bring significant improvements. When using linguistic data, feature selection yielded the best overall mean improvement (across all train-test pairs) of 4%. Among the domain adaptation methods, the greatest improvement of 3% was achieved by subspace alignment.

Keywords: Fake news; Psycholinguistics; Domain adaptation; Data set shift

1. Introduction

Recognizing fake news is centered on the automatic detection of intentionally misleading false news stories (Conroy, Rubin, and Chen 2015; Allcott and Gentzkow 2017). The problem has grown in importance because of changes in the dissemination of information. Traditional news publishers no longer control the distribution of news; information circulates among Internet users at a fast pace owing to the rise of social networks. Once communicated via social media, inaccurate, distorted, or false information is amplified and has a tremendous potential for large-scale, real-world consequences (Vosoughi, Roy, and Aral 2018).

Research on fake news detection has been especially lively since the 2016 US presidential campaign. This was when the New York Times defined fake news as “a made-up story with an intention to deceive”.^a Some researchers and journalists hypothesized that fake news had a real influence on voter preference during that campaign (Allcott and Gentzkow 2017). According to a recent Pew Research Center study from mid 2019, Americans rate fake news as a larger problem

^a<https://www.nytimes.com/2016/12/06/us/fake-news-partisan-republican-democrat.html>.

than racism, climate change, and terrorism.^b Given the above context, establishing methods for fact-checking the veracity of information presented through social media is critical.

The most reliable approach is without a doubt human fact-checking. Unfortunately, it is also the most expensive and difficult solution in terms of labor costs and time needed to find and process reliable sources of information, which are often unstructured. Moreover, not all information needed for checking facts is readily available. For example, when verifying certain claims it may be necessary to have access to expert knowledge contained in scientific databases or which is otherwise publicly unavailable. Due to these reasons and also the sheer amount of information that is generated and shared online, human verification is realistically applicable only to a fraction of news and claims.

Accordingly, many researchers and companies have turned towards investigating automated methods of fake news detection. Firstly, this is reflected by the rise of several start-ups dedicated to fake news detection technology, later acquired by the main players in the social network industry. Secondly, the surging interest in automated fake news recognition has resulted in an increasing amount of research on this topic.

Studies on the methods of automated fake news detection typically fall into the following categories: content-based (detection based solely on text), source-based (based on features of the source), and diffusion-based (based on patterns of spreading the news).

In this article, our main focus is on content-based methods. Namely, we are interested in techniques that predict whether a particular document—usually a news article—is fake or not, using that document as the only source of information. Predicting its veracity is carried out on the level of content, syntax, style, and other implicit features (Newman *et al.* 2003) and is based on selected methods available in the field of natural language processing, especially in the area of deep neural networks. We investigated multiple fake news data sets generated by using different assumptions to see if universal features can be identified and generalized to widely applicable models. We compared the content-based approach to a chosen state-of-the-art automated fact-checking system that uses Wikipedia as its knowledge base.

The definition of fake news remains problematic in the context of research. It is not sufficient to say that it is news that contains false information. The key feature is that this kind of article is also intentionally fabricated (Conroy *et al.* 2015; Allcott and Gentzkow 2017), and we follow this understanding in our article.

This often entails the use of slightly different language and stylistic meanings (Newman *et al.* 2003; Rashkin *et al.* 2017; Levi *et al.* 2019). Moreover, we limited our study to manually written content. We did not focus on recognition of AI-generated (neural) fake news, which may soon be a serious danger due to the rapid improvements in automated content generation tools. However, recent studies have revealed that although AI-generated news might be recognized by humans as more-trustworthy, the accuracy in differentiating between human written and AI-generated texts reaches even 92% (Zellers *et al.* 2019).

Content-based approaches make it possible to avoid tedious fact-checking procedures and does not require collections of truthful documents against which to compare the input claim.^c Our method is therefore more resource-efficient, but at the same time more difficult. Since our method does not require any additional resources beyond the text that is being analyzed, it has a potential to be more universally applicable than other approaches such as fact-checking^d

Our paper is organized as follows. Section 2 presents existing work on the topic of fake news detection; Section 3 introduces the data sets used in our experiments. Section 4 describes feature spaces used for representing texts as well as baseline classification methods used in two scenarios:

^b<https://www.journalism.org/2019/06/05/many-americans-say-made-up-news-is-a-critical-problem-that-needs-to-be-fixed/>.

^cAs, for example, Fever competitions; see: <http://fever.ai>.

^dHowever, it is likely to fail with sentences like “NHE grew 4.6% to \$3.8 trillion in 2019, or \$11,582 per person, and accounted for 17.7% of Gross Domestic Product (GDP).” as it can be evaluated only by fact-checking.

Section 4.4 describes the performance of models trained and tested on the same data set (or in-domain), while Section 4.5 refers to a scenario in which the models were trained and tested on different data sets (or cross-domain). The high accuracies obtained when training and testing on the same data set can wrongly indicate that the problem of automated fake news detection is nearly solved. Testing the model on a different data set than the training one showed a markedly different picture. In this scenario prediction quality decreased significantly, which demonstrates that creating a universally applicable model for fake news detection is far from easy.

The core part of our paper is focused on possible methods of dealing with this quality decrease. We tested five broad categories of methods:

- A leave-one-out approach described in Section 5. In these experiments, we tested the effect of training on several data sets.
- Automated fact-checking described in Section 6. This method is based on two stages: retrieval of the relevant article (evidence) from Wikipedia and inference about the veracity of the input text (claim).
- Feature selection, which we investigated in Section 7. Section 7.1 describes results obtained from finding features that are relevant to not only one but also many data sets, and therefore potentially more universal (feature intersection), and Section 7.2 presents a classifier on such a feature space.
- Methods from the area of machine learning related to the concept of data set shift in Section 8. Data set shift techniques deal with differences between the distributions of training and testing data that may lead to unreliable predictions. The approaches include instance reweighting and common representation space methods such as subspace alignment and geodesic flow kernel.
- Deep learning approaches to domain adaptation are presented in Section 9. The scenario that we are referring to is also called transductive transfer learning and should be distinguished from unsupervised and inductive transfer learning. The latter two are more common in Natural Language Processing (NLP), with typical examples being pretraining and fine-tuning of transformer-based language models.^e

We summarize all the results and conclude the paper in Section 10.

2. Related work

Numerous approaches have been proposed to solve the problem of fake news detection. The following review will focus solely on text-based methods: analyzing textual content to predict the veracity of an input text. Broadly speaking, the approaches can be classified either as knowledge-rich (fact-checking) or knowledge-lean.

2.1 Fact-checking

Fact-checking is based on the confirmation or rejection of claims made in a text piece with the use of explicit references to knowledge sources, ideally credible ones such as Wikipedia (Thorne *et al.* 2018). Automated fact-checking is done in a two-step procedure: in the first step relevant articles are retrieved from the knowledge base, and in the second step, inference (refuting or supporting the input claim) is performed by a neural network.

The approaches to fact-checking go beyond Wikipedia: Wadden *et al.* (2020) introduce scientific claim verification and demonstrate that domain adaptation techniques improve performance compared to models trained on Wikipedia or political news. In Augenstein *et al.* (2019), the

^ehttps://en.wikipedia.org/wiki/Domain_adaptation.

authors use a data set from fact checking websites and apply evidence ranking to improve veracity prediction. The research by Leippold and Diggelmann (2020) introduces a data set for verification of climate change-related claims and adapts the methodology of Thorne *et al.* (2018). Another study (Rashkin *et al.* 2017) probes the feasibility of automatic political fact checking and concludes that stylistic cues can help determine the truthfulness of text.

Fact-checking and fake news detection have been the main topics of CLEF competitions since 2018. In the 2018 edition, the second task “Assessing the veracity of claims” asked to assess whether a given check-worthy claim made by a politician in the context of a debate/speech is factually true, half-true, or false (Nakov *et al.* 2018). The data set consists of less than 90 verified statements from a presidential debate. In 2019 (Elsayed *et al.* 2019), “Task 2 Evidence and Factuality” asked to (A) rank a given set of web pages with respect to a check-worthy claim based on their usefulness for fact-checking that claim, (B) classify these same web pages according to their degree of usefulness for fact-checking the target claim, (C) identify useful passages from these pages, and (D) use the useful pages to predict the claim’s factuality. In CLEF 2020 (Barrón-Cedeño *et al.* 2020), “Task 4: Claim Verification” asked to predict the veracity of a target tweet’s claim by using a set of web pages and potentially useful snippets contained within. The data sets for the 2019 and 2020 tasks are in Arabic. The CLEF 2021 “Task 3a: Multi-Class Fake News Detection of News Articles” (Nakov *et al.* 2021b) focused on multi-class fake news detection and topical domain detection of news articles in several languages, including English. The data are not publicly available. The CLEF 2021 “Task 2 Detecting previously fact-checked claims from tweets” used data from Snopes and ClaimsKG (Tchechmedjiev *et al.* 2019) to rank previously fact-checked claims in order to measure their usefulness.

The fully automated fact-checking pipeline could include steps such as identifying claims worthy of fact-checking, followed by detecting relevant previously checked claims and then only focus on verification of selected, check-worthy claims (Nakov *et al.* 2021b). Systems like this can facilitate the work of fact-checking professionals (Nakov *et al.* 2021a).

2.2 Knowledge lean

The second type of methods infer veracity only from the input text and do not use any external sources of information. Such approaches are based on various linguistic and stylometric features aided by machine learning.

Features such as n-grams, LIWC psycholinguistic lexicon (Pennebaker *et al.* 2015), readability measures, and syntax were all used in Pérez-Rosas *et al.* (2018). Then, an SVM classifier was applied to predict veracity. A similar set of features including n-grams, parts of speech, readability scores, and the General Inquirer lexicon features (Stone *et al.* 1966) was used in Potthast *et al.* (2018). Interestingly, they argue that style-based fake news classification does not generalize well between news of both partisan orientations of the USA. A combination of features at four levels—lexicon, syntax, semantic, and discourse—was used in Zhou *et al.* (2020). They applied classification methods such as SVM (with linear kernel), random forest (RF), and XGBoost. In yet another study, (Przybyla 2020) designed two classifiers: a neural network and a model based on stylometric features to capture the typical fake news style. Style analysis allowed to extract sensational and affective vocabulary that is typical of fake news.

Performance drops in different domains were carefully studied by Silva *et al.* (2021) and concluded with a new framework that jointly preserves domain-specific and cross-domain knowledge. This allowed for a new cross-domain classifier to be trained on data selected by an unsupervised technique (and manually labelled).

The problem of excessive dependence on training sets and the low robustness of text-based fake news detection was observed by Janicka *et al.* (2019). The authors used four different fake news corpora to train models in both in-domain and cross-domain settings. They concluded that the results achieved by models trained and tested on the same corpus (in-domain) are unrealistically

Table 1. Summary of data sets

Data set	Size	Average number of sentence	Average number of characters	Domains
Kaggle	10,384 true	42.14 ± 39.99	5215 ± 4314	Politics
	9781 fake	28.70 ± 50.44	3964 ± 5710	
LIAR	2063 true	1.16 ± 0.50	105 ± 43	Politics
	2466 mostly-true	1.16 ± 0.52	108 ± 45	
	2638 half-true	1.19 ± 0.58	112 ± 47	
	2108 barely-true	1.16 ± 0.50	108 ± 46	
	2511 false	1.13 ± 0.44	100 ± 44	
	1050 pants-fire	1.18 ± 0.54	103 ± 42	
AMT	240 true	4.49 ± 2.82	699 ± 179	Politics, technology, business, education, sport, entertainment
	240 false	5.25 ± 1.96	662 ± 204	
FakeNewsNet	91 true	25.26 ± 29.26	3593 ± 4358	Politics, business, breaking news, local news, medicine, race
	91 fake	21.95 ± 12.85	2737 ± 1704	
ISOT	21,417 true	14.82 ± 11.37	2383 ± 1685	Politics
	23,481 false	14.91 ± 13.78	2547 ± 2532	

optimistic with regard to the true performance of the models on real-world data. The performance in the cross-domain setting was on average 20% worse than in the case of the in-domain setting. Compared to Janicka *et al.* (2019), our work is a further step forward: our goal is to find the means to improve the robustness and cross-domain performance of fake news detection models.

3. Data sets

As our work tried to assess a universal tool for fake news recognition, we used several publicly available data sources. The available data sets vary in text length, origin, time span, and the way of defining truthfulness of information. In all cases, the annotation was performed at an article level. The key feature of fake news is that it is an intentionally written disinformation. The classification of the collected data as fake or true was not based on the number/ratio of misleading claims/sentences. Articles classified as fake were often a mixture of true and false sentences. The ways of ascribing news veracity vary among the data sets. The following strategies were used: (1) labelling based on the source reliability (the content was not analyzed), (2) manual verification of text by experts, and (3) generation of synthetic data (only fake news) by means of crowdsourcing. Hence, unintentional inaccuracies in a text did not result in it being classified as fake. A summary of the investigated data sets can be found in Table 1, while samples of texts are provided in Table 2.

In our studies we used the following data sets:

Table 2. Examples of news in the investigated data sets**LIAR**

Hillary Clinton has taken over \$800,000 from lobbyists.

Close to 30% of our federal prison population consists of illegal immigrants.

AMT, *Ford to invest \$1.2bn in Michigan plants*

Ford has said it will spend \$1.2bn (£1bn) as part of a planned upgrade of three Michigan plants. It said \$850m will be spent on retooling its factory in Wayne where Ford plans to build Bronco and Ranger models. (. . .)

US President Donald Trump, who put pressure on Ford over its planned Mexico investment, tweeted earlier on Tuesday: “Big announcement by Ford today. Major investment to be made in three Michigan plants. Car companies coming back to US. JOBS! JOBS! JOBS!” Ford’s US investment announcement is largely in line with a previous agreement it reached with the United Auto Workers union.

Kaggle, *Whoopi: Are the Trump Administration Values ‘Really Much Different than the Taliban’s?’ - Breitbart*

Tuesday on ABC’s “The View,” Whoopi Goldberg wondered if the “values” of Trump administration are “really much different than the Taliban’s?” Goldberg said, “We also keep hearing about terrorists hating our American values. We had this conversation yesterday. So let me ask you now — now, we have had a leader who’s repeatedly demeaned women, wants to defund organizations that benefit woman, calling on the media to shut up, specifically wants to give preferential treatment based on religion, are these values really much different than the Taliban’s?” When asked if she was talking about Trump’s values, Goldberg continued, “Well no, the values that we are listening to. One of the things that you read yesterday was the piece of the . . . the language in the ban, which was about if you disrespect women. (. . .) My question is, what’s happening?” Follow Pam Key on Twitter @pamkeyNEN

ISOT, *Bad News For Trump — Mitch McConnell Says No To Repealing Obamacare In 2018*

Republicans have had 7 years to come up with a viable replacement for Obamacare but they failed miserably. After taking a victory lap for gifting the wealthy with a tax break on Wednesday, Donald Trump looked at the cameras and said, We have essentially repealed Obamacare and we will come up with something that will be much better. Obamacare has been repealed in this bill, he added. Well, like most things Trump says, that’s just not true. But, if the former reality show star could have done that in order to eradicate former President Obama’s signature legislation, he would have and without offering an alternative. Senate Majority Leader Mitch McConnell told NPR that “This has not been a very bipartisan year.” We still recall him saying that his number one priority is making sure president Obama’s a one-term president. Well, we’re hoping that Trump doesn’t last a full term. Funny how that works. Photo by Chip Somodevilla/Getty Images

FNN (FakeNewsNet), *Clinton’s Exploited Haiti Earthquake ‘to Steal Billions of Dollars from the Sick and Starving’ - Freedom Daily*

0 SHARES Facebook Twitter Bernard Sansaricq, former president of the Haitian Senate, issued a blistering statement condemning the Clinton Foundation, which has been posted at Donald Trump’s campaign website. (. . .)

- (1) **Kaggle Fake News Data set**^f – consists of 20.8 k full-length news texts labeled as either fake or true. The fake portion of data^g was collected from websites tagged as unreliable by the BS Detector Chrome extension created by Daniel Sieradski. The fake news mainly stems from the US from 1 month in 2016 around the elections. Since we found some non-English texts in this database, we used the *langdetect* library^h to filter them out. As a result, we ended up with 20,165 samples: 10,384 true and 9781 fake.
- (2) **LIAR** (Wang 2017) – 12.8 k short statements categorized into six categories (*pants-fire*, *false*, *barely-true*, *half-true*, *mostly-true*, and *true*). The data come from politifact.com and were manually annotated by experts not only for truthfulness, but also for subject, context, and speaker. The statements were collected primarily in the period from 2007 to 2016

^f<https://www.kaggle.com/c/fake-news/data>.

^g<https://www.kaggle.com/mrirdal/fake-news>.

^h<https://pypi.org/project/langdetect/>.

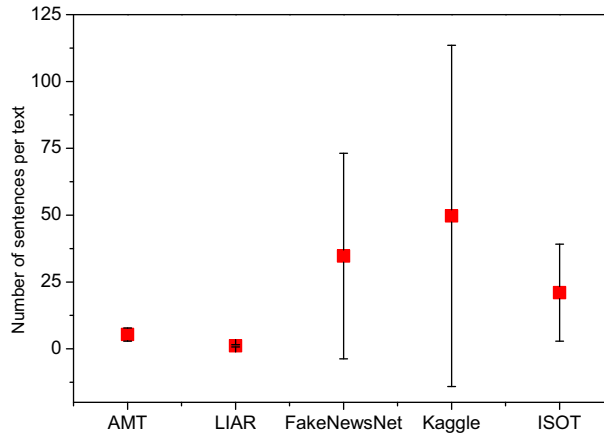


Figure 1. Mean and standard deviation of the number of sentences in each corpus.

from various sources (e.g. news releases, radio interviews, tweets, campaign speeches) and cover diverse topics such as elections, education, healthcare, etc. It is worth noting that the statements often contain numerical values, for example “Close to 30% of our federal prison population consists of illegal immigrants.” Moreover, some of the claims require broader context for their verification. For instance, the claim “Crime is rising” does not provide detailed information about location and time period. This data set was originally divided into three parts—train, validation, and test—which we merged. In our research, we decided to use only texts annotated as *pants-fire*, *false*, or *true*. Finally, we ended up with 5624 samples: 3561 fake and 2063 true.

- (3) **AMT** (Pérez-Rosas *et al.* 2018)—consists of 240 true and 240 false news articles. The true portion of the data was collected from US news web sites such as ABCNews, CNN, USAToday, NewYorkTimes, FoxNews, etc. and covers six domains: sports, business, entertainment, politics, technology, and education. The fake part was created based on real news via Amazon Mechanical Turk. Each fake news story was written with the aim to imitate the topic and style of a true article, but at the same time presenting fake information.
- (4) **FakeNewsNet** (Shu *et al.* 2020) – a data set created by using FakeNewsNet resources and tools with help from politifact.com as a source of information about the veracity of individual texts. It contains the most popular fake news stories that were spreading on Facebook in 2016 and 2017, balanced with some reliable texts. These fall into several categories such as politics, medicine, and business.
- (5) **ISOT Fake News Data set** (Ahmed, Traore, and Saad 2017) – the data set consists of 21,417 real news items collected from reuters.com and 23,481 news items obtained from sites flagged as unreliable by Politifact and Wikipedia. The collected articles are from 2016 to 2017 and mainly deal with politics and world news.

We can see that the length and style of news differs between data sets. As the LIAR data set comprises data from a fact-checking portal, it contains short claims that often express a single fact that needs to be verified. Figure 1 illustrates the mean and standard deviation of the number of sentences for each data set. As we can see, LIAR texts are on average the shortest among the data from all the above data sets. We can also observe that the AMT data have a relatively small standard deviation in text length, which can be ascribed to the data preparation process where only article fragments of a given size were considered. The Kaggle data set shows the biggest spread in text length, and at the same time the texts are on average the longest among all data. In most data

Table 3. Summary of other results on given data sets

Data set	Article	Method	Accuracy
Kaggle	Leroy Todd ^a	Not available	0.99
	Matt Gallagher	Not available	0.98
	Matthew LeGate	Not available	0.97
LIAR (6 classes)	Wang (2017)	CNN + embeddings	0.27
	Popat <i>et al.</i> (2018)	LSTM + embeddings	0.26
AMT	Pérez-Rosas <i>et al.</i> (2018)	SVM + linguistic features	0.73
FNN	Zhou <i>et al.</i> (2020)	XGBoost + linguistic features	0.89
ISOT	Ahmed <i>et al.</i> (2017)	LSTM + TF-IDF	0.92

^awww.kaggle.com/c/fake-news/leaderboard.

sets, there is some overlap when it comes to the time span of the origin of the collected texts. We do not have information about AMT time span, but every other data set covers at least some part of the year 2016, with the Kaggle time window being the narrowest at 1 month and LIAR being the broadest at 10 years. It was the year of the presidential election in the United States, which brought greater social awareness of issues connected to the spread of fake news and encouraged researchers to devote more attention to this problem. However, the topics covered in each data set are not limited to politics, but also include other social issues, with some differences in the proposed categorization of the topics.

To identify patterns in the data sets that the models might exploit, we computed the mutual information between bigrams in the text and the labels according to the method described by Schuster *et al.* (2019). Popular bigrams associated with each of the two labels are the names of politicians from both parties. However, there is no clear attribution. Moreover, even in the same data set, different variants may be associated with opposite classes. For example, Trump-related bigrams are associated with true texts in Kaggle (except “Donald Trump”, as this one is linked to fake texts) and AMT, but with the fake class in ISOT (“President Trump”). Clinton-related bigrams are associated with the fake class in Kaggle (except “Mrs Clinton”, which is linked to the true class), FNN, and ISOT. Other bigrams with high scores include social issues (LIAR) or month names (Kaggle). This indicates that even the bigrams most correlated with the true and false classes have limited utility for cross-domain classification.

The highest classification accuracies previously reported for selected fake news data sets are listed in Table 3. The results presented there are not directly comparable to ours as we used cross validation instead of splitting the data into train and test sets.

4. Fake news detection methods

In our work, we define fake news recognition as a text classification task, which reduces the scope of investigation to methods connected with news texts only. Images, sources, authorship, and other features of the news are outside the scope of our analysis. In this section, we introduce various sets of features used to represent texts in further experiments. Then, we describe the application of several methods from the fields of machine learning and deep learning that we used to predict text veracity. The methods applied and described in this section are generic, meaning that no adaptation was introduced to increase their robustness in the cross-domain scenario. Such adaptations are the subject of the subsequent Sections 8 and 9.

Table 4. Dictionaries of hedges and exclusions

Exclusion	apart from, aside from, besides, but, exclude, excluding, except (for/that), excepting (that), if not, unless, without, with the exception of
Hedges	somehow, somewhat, sort of, kind of, may, might, possibly, presumably, apparently, possible, probably, probable, appear(s/ed) to be, seem(s/ed) to be, suggest(s/ed/ing), it may/might be, to some extent, tend(s) to, may suggest that, wish to suggest, has/have been claimed that, assume(d/s/ing), relatively, allegedly, supposedly, evidently

4.1 Feature space

- (1) **Bag-of-words (BOW)** – in the first approach, for each data set we calculated the term frequency–inverse document frequency (TF-IDF) of uni-, bi-, and tri-grams.
- (2) **Linguistic features** – using various sources of information, we created a set of 271 features containing semantic, syntactic, and psycholinguistic information. More than 180 features stem from the General Inquirer (Stone *et al.* 1966), a tool providing a wide range of psycholinguistic categories for analyzing text content in terms of sentiment, emotions, as well as multiple other social and cognitive categories. We enriched the data with hand-crafted dictionaries of linguistic hedges and exclusion terms, which are presented in Table 4. Each feature was normalized by the number of all words in the text. Similarly, using the spaCy library,ⁱ we added part-of-speech information in each text. We also added the sentence-level subjectivity measure from an LSTM-based model^j that was trained on a data set released by Pang and Lee (2004). Finally, various readability indicators were used: Flesh-Kincaid (Flesch 1948; Kincaid *et al.* 1975), ARI (Smith and Senter 1967), Coleman-Liau (Coleman and Liau 1975), etc.
- (3) **GloVe** (Pennington, Socher, and Manning 2014) – we used 100-dimensional word embeddings that were pre-trained using aggregated global word co-occurrence statistics found on Wikipedia and the Gigaword corpora.
- (4) **Universal Sentence Encoder** (Cer *et al.* 2018) – a versatile sentence embedding model based on the Transformer architecture trained in a transfer learning manner. It converts text into 512-dimensional vector representations, which can capture rich semantic information. They can be used in various downstream tasks and have proven to yield good results in many text classification problems.
- (5) **ELMo embeddings** (Peters *et al.* 2018) – we used this model to represent each text by a 1024-dimensional vector. These features were used exclusively for visualization, to compare whether they convey more information than the embeddings obtained from the Universal Sentence Encoder.

4.2 Models

As the baseline approach, we used a support vector machine classifier with linear kernel trained on bag-of-words features. The default parameters were used: $C=1$ and squared hinge loss as the loss function.

To create a classifier based on linguistic features, we tested various machine learning algorithms from the *scikit-learn* (Pedregosa *et al.* 2011) library, such as support vector machines, stochastic gradient descent, extra trees, and gradient-boosted trees (XGBoost). The default hyperparameter

ⁱ<https://spacy.io>.

^jhttps://github.com/fractalego/subjectivity_classifier.

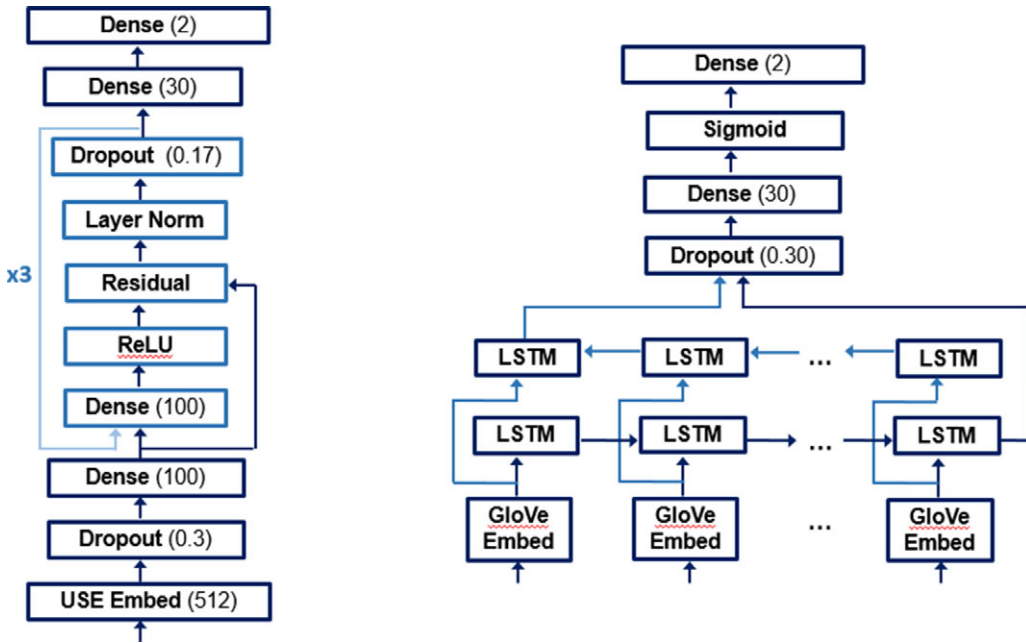


Figure 2. The architecture of two models: based on USE embeddings (left) and bi-directional LSTM using GloVe embeddings (right).

values were used in each case. The best test set performance for all five data sets was obtained by XGBoost, which was used in further experiments.

Next, we tested a bidirectional-LSTM architecture initialized with 100-dimensional GloVe embeddings. The scheme is shown in Figure 2 on the right. According to a benchmark study on fake news detection methods by Khan *et al.* (2021), this classifier achieved the best results among the tested approaches. Both the output dimension of the LSTM and the number of time steps were set to 100. The outputs of the last LSTM units (one for each direction) were concatenated and followed by dropout and a dense classification layer with the sigmoid activation function. We used the ADAM optimizer to minimize the binary cross-entropy loss. The model was implemented using the PyTorch library.

As a basic model utilizing USE embeddings, we used the LinearSVC classifier which takes as input a vector representation of whole texts. This approach proved to be very successful in some text classification tasks with limited data thanks to SVM’s robustness and regularization (Xu, Caramanis, and Mannor 2009). Since one of our goals was to compare different deep domain adaptation methods, we also investigated the performance of a simple neural model utilizing USE embeddings as a baseline. The architecture is presented in Figure 2 (left).

The embeddings were used as the input and were followed by three blocks, each constituted of a dense layer with residual connection and a normalization layer. Finally, the obtained representation was projected to two classes by two dense layers. Hyperparameters such as dropout and dense layer dimensions were fine-tuned using the Kaggle data set, while the epoch count was set to 200 based on the mean scores obtained for all data sets.

Moreover, we tested the BERT language model (Devlin *et al.* 2019), which is based on the Transformer architecture and has achieved state-of-the-art results in many NLP tasks. We used the TensorFlow implementation and tested sequence lengths of 32, 128, and 512. The number of training epochs was limited to 5, as there was no further improvement in the classification performance. Typically, one or two epochs provided maximum accuracy. Therefore, we finally used a sequence length of 128 and trained the model for 2 epochs.

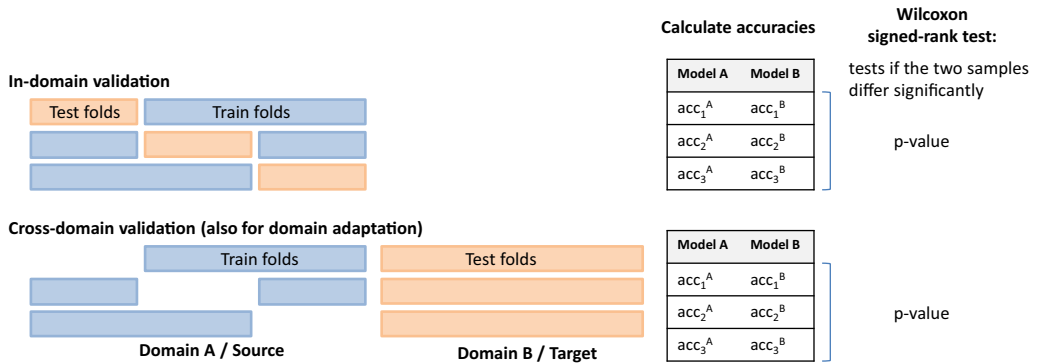


Figure 3. The scheme of validation and comparing machine learning models under different conditions: in-domain and cross-domain setting.

4.3 Evaluation of classifiers

Comparison of classifier performance is a difficult task and cannot be limited to the analysis of the main score, often calculated using *k*-fold cross validation. Although the model with the best mean performance is expected to be better, this is not always true. Such a simple approach can be misleading, as the difference in performance might be caused by chance. Therefore, to be sure that one model provides significantly higher accuracy than the others, statistical tests should be performed. Fortunately, there is a variety of statistical methods enabling the selection of best performing machine learning model.

As the first step we decided to use a Cochran’s Q test (Cochran 1950), which is used to determine if there are differences on a dichotomous dependent variable (which can take only one of two possible values) between more than two related groups. It tests the null hypothesis that the proportion of “successes” is the same in all groups. We applied this test to the predictions of the investigated models. If the null hypothesis was rejected, multiple pairwise comparisons between groups were performed as the second step.

We applied the Wilcoxon signed-ranked test (Wilcoxon 1945; Demšar 2006) to verify if two samples differ significantly from each other, testing the null hypothesis that the related paired samples are drawn from the same distribution. It is a non-parametric version of the paired Student’s t-test. Contrary to Student’s t-test, the Wilcoxon signed-rank pair test does not require normally distributed values.

Figure 3 presents how the pair-wise evaluation and comparison of models under different settings were performed. Instead of using raw models predictions like in Cochran’s Q test, we compared the accuracies obtained during *k*-fold cross validation. In our experiments, we used 10-fold cross validation and set the significance level to 0.05. In the results tables, we marked the groups of models for which the Cochran’s Q test revealed no differences with a gray background. Moreover, within the remaining groups, the pairs of models for which the Wilcoxon signed-ranked test indicated a *p*-value > 0.05 (non-significant differences) are marked with the same letters in the superscript.

4.4 In-domain results

We compared the performance of the above classifiers in data set-specific or in-domain settings using 10-fold cross validation. The obtained results are presented in Table 5. In general, all classifiers performed significantly better on the Kaggle and ISOT data sets when compared with the others. This fact does not seem surprising considering that those collections are several times larger than the others.

Table 5. In-domain accuracy of different fake news detection models on five data sets

	Kaggle	ISOT	AMT	LIAR	FNN
LinearSVC + BOW	0.97	0.99	0.40 ^a	0.67	0.75
XGBoost + ling. feat.	0.99	0.98	0.80	0.63	0.70
LinearSVC + USE	0.93	0.99	0.57	0.68	0.72
Dense + USE	0.92	0.99	0.51	0.68	0.74
Bi-LSTM + GloVe	0.91	0.99	0.56	0.63	0.73
BERT	0.98	0.99	0.67	0.66	0.71

^aWe could not replicate the results of Perez-Rosas et al., who reported an accuracy of 0.62 using similar settings: a linear SVM classifier based on TF-IDF values for unigrams and bigrams with 5-fold cross-validation; with these settings we achieved an accuracy of 0.43. The accuracy is significantly higher for char-level settings - 0.66. The value reported here is for 10-fold cross-validation of an SVM classifier based on TF-IDF values for unigrams, bigrams, and trigrams.

Taking into account the results on all data sets, the best-performing classifier is XGBoost trained on linguistic features. It achieved the highest accuracy on two out of five data sets: Kaggle and AMT. Moreover, its accuracy on the three remaining data sets—ISOT, LIAR, and FNN—was lower than the accuracy of the best-performing classifiers by only 0.01, 0.05, and 0.05, respectively. Finally, its advantage over other models on the AMT data set is enormous: the obtained accuracy is higher by 0.13 when compared to the runner-up. It seems that this data set is the most challenging for the other methods. The accuracies of models using USE embedding are very low: 0.57 obtained by LinearSVC and 0.51 achieved by a neural network based on dense layers. The poor performance of classifiers based on embeddings in the case of the AMT data set is discussed further. Even worse results were provided by LinearSVC using bag-of-words features (0.40). The reason behind such poor results is strong overfitting to the training data. In 10-fold cross validation settings, the accuracy on the training sets regularly reached a value of almost 1.0, while on the test parts it fluctuated around 0.40. In the case of char-level settings (using character-level n-grams), the values are respectively 0.82 and 0.66, but we decided to focus on word-level analysis. The comparison of our results with the best existing ones (as in Table 3) revealed that we managed to outperform the accuracy on the ISOT data set (0.92 vs. 0.99) reported in Ahmed *et al.* (2017). Our results are similar in the case of Kaggle; however, we did not obtain a higher accuracy than Zhou *et al.* (2020) for the FNN data set. As for the LIAR data set, we obtained significantly higher accuracy than the best results reported by Wang (2017), Popat *et al.* (2018). However, these values cannot be directly compared. Six classes were used in the original analysis, while we narrowed it down to two. It should be noted that we used the same method for all data sets (without hyperparameter tuning for a specific data set). Hence, in our opinion, the results are satisfactory.

We also investigated which linguistic features are the most important for classifying fake news. The results are shown in Table 6 and indicate that among the linguistic features, part-of-speech tags are the most informative. The addition of other linguistic features (the first row of Table 6) did not significantly improve the accuracy—only by 0.03 for AMT and 0.04 for FNN. In the case of LIAR data set, it even led to a slight decrease in performance.

The obtained results indicate that designing a high-accuracy classifier is more challenging for some data sets than for others. To get insight into the arrangement of data in both high-dimensional feature spaces—USE embeddings (512) and linguistic features (271)—we applied t-Distributed Stochastic Neighbor Embedding (t-SNE). The vectors containing linguistic features were standardized before dimensionality reduction. Figure 4 presents the obtained results for three selected data sets.

This visualization can explain why the classification based on USE embeddings failed in the case of AMT. For a large number of points related to true texts, points related to fake texts occurred in

Table 6. In-domain accuracy of different fake news detection models on five data sets for various linguistic features

	Kaggle	ISOT	AMT	LIAR	FNN
XGBoost + ling. feat.	0.99	0.98	0.80	0.63	0.70
XGBoost + GI	0.79	0.88	0.53	0.64	0.70
XGBoost + POS	0.98	0.97	0.77	0.66	0.66
XGBoost + subjectivity	0.61	0.66	0.52	0.63	0.55
XGBoost + readability	0.95	0.84	0.66	0.63	0.61

a very close distance. This might be caused by the way in which this data set was created. Initially, legitimate news pieces were collected from a variety of websites (their veracity was confirmed by manual fact-checking). However, the fake news collection did not originate in any existing source, but was written specifically for this data set. Amazon Mechanical Turk workers were asked to write a fake version for every legitimate news story. At the same time, they were requested to imitate the journalistic style and preserve the names mentioned in the original news. As a result, both the originals and their fake counterparts were closely related. Therefore, their USE embeddings were often similar (USE embeddings proved to be successful in finding semantic similarity).

For a comparison, we also visualized more expressive embeddings—ELMo (Figure 4b). The t-SNE plots for both USE and ELMo embeddings appear quite similar. Still, the ELMo representations for paired true and fake texts were often located close to each other. However, in the case of linguistic representation the samples in AMT data sets were much more scattered, that is vectors for related original and fabricated texts were more separated. It indicates that USE and ELMo embeddings are more related to the content of the text (its semantics), while the linguistic features are not strongly related to its meaning but contain information about part-of-speech tags, readability indices, etc. Even if both texts cover the same topic they are written differently, which results in a larger change in linguistic features than their embeddings. Therefore, the linguistic feature vectors of the fake and true texts are better separated, which allows for more accurate classification.

In the case of LIAR, linguistic representations of the data tended to form many well-separated clusters, while USE embeddings seemed more dispersed. This might be due to the length of the texts in the LIAR data set. As they contain single short sentences, their linguistic representations contain many zeros. Unfortunately, true and fake texts do not form separate clusters, which makes classification very difficult. The visualization of the data arrangement suggests that USE embeddings might work better for this data set.

The distributions of USE embeddings and linguistic features obtained for the Kaggle data differed significantly. Most strikingly, the linguistic representations of fake news formed one large cluster and a few smaller ones. As for USE embeddings, they formed smaller clusters. The points related to both classes rarely appeared in the same clusters. The classifiers utilizing USE embeddings reached an accuracy of 0.93. An even higher accuracy of 0.99 was achieved when linguistic representation was used.

To conclude, the use of linguistic and psycholinguistic features proved to be the most versatile approach for the detection of fake news. However, it only works for longer texts and not for very short ones, like in the LIAR data set.

As the t-SNE plots revealed large differences between the studied data sets, we decided to see how the classification accuracy relates to the number of training samples.

The goal of this analysis was to show that variability in classification accuracy was not solely a result of different data set sizes. We also hoped to observe how quickly the classification models

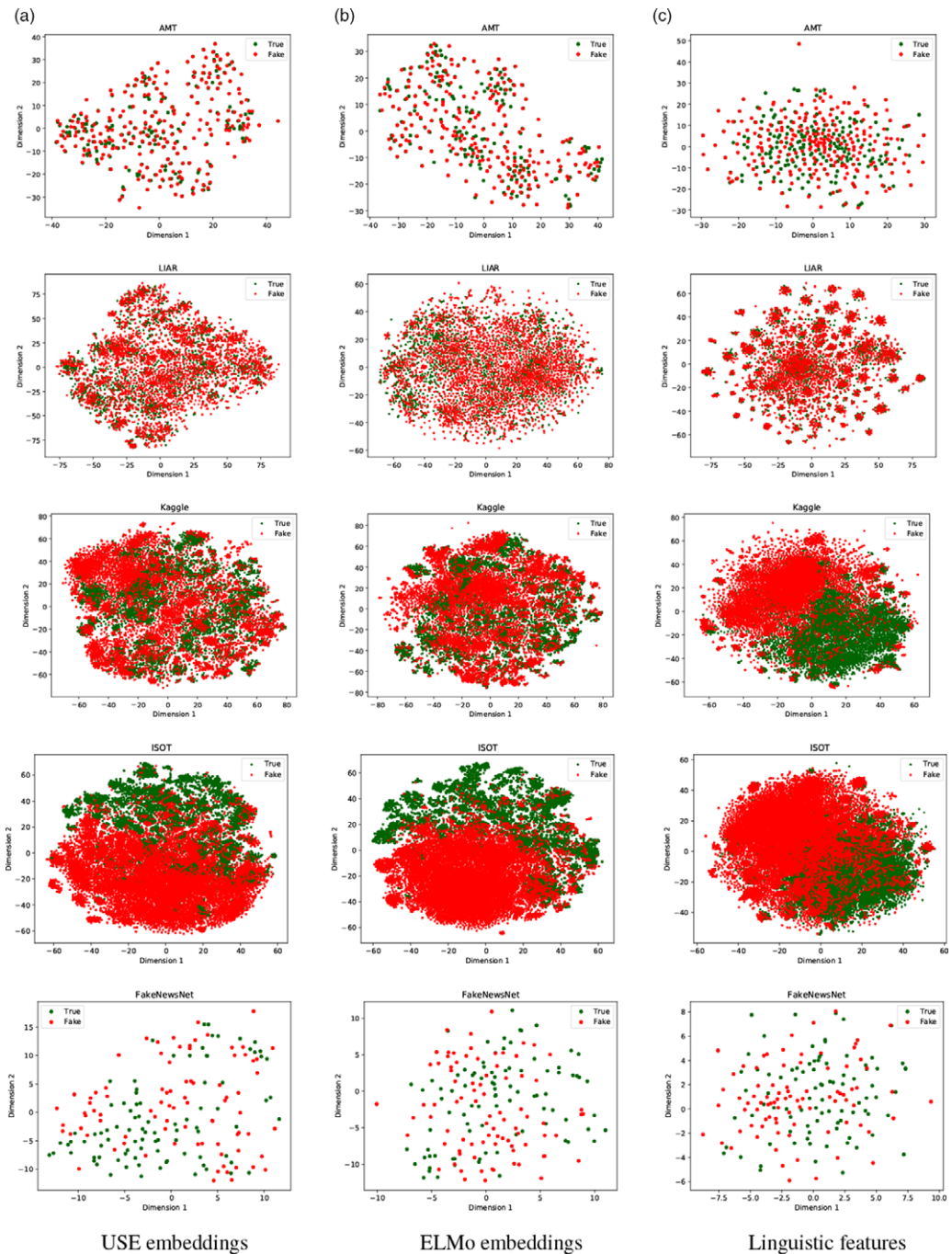


Figure 4. t-SNE visualization of USE embeddings, ELMo embeddings, and linguistic features.

learn and to approximate how accuracy increases when adding training samples. Lastly, the behavior of classifiers with limited ground truth access was examined.

The classifiers were trained on randomly selected samples that maintained class representations. The remaining samples were used as the test set. Due to large variance in the

results, especially for a small number of training samples, the whole procedure was repeated 10 times and the obtained accuracy scores were averaged.

Figure 5 shows the results obtained for two classifiers: LinearSVC and XGBoost. We compared two representations: USE embeddings and linguistic features. Interestingly, the obtained plots reveal significant differences. The most expected are the plots for Kaggle and ISOT: the larger the training set, the higher the accuracy accompanied by lower variance. As it turned out, LinearSVC performed slightly better than XGBoost when trained on USE embeddings. The behavior of models trained on FakeNewsNet was very similar to the classifiers trained on Kaggle and ISOT. Unfortunately, due to the much smaller size of FakeNewsNet, it could not be compared in larger ranges. Nevertheless, the observed similarity suggests that the addition of new labeled samples would improve classification accuracy.

The plots obtained for the AMT data set look strikingly different. The most surprising is the behavior of the XGBoost classifier trained on USE embeddings. Its accuracy did not exceed 0.5 and decreases with the increasing size of the training set. Such unusual behavior occurs exclusively for the AMT data set, which indicates that it is related to the nature of USE embeddings of texts contained in that data set (as previously discussed). t-SNE visualization of these embeddings (Figure 4) shows that they often occur in highly similar pairs: the embeddings of the original (true) text and the fake one. As a reminder, the USE embeddings for original and fabricated texts are often close to each other, as they cover the same topic. Interestingly, LinearSVC deals significantly better with such specific data.^k The situation was different in the case of classifiers trained on linguistic features: the accuracy for both LinearSVC and XGBoost increased upon expansion of the training set for the AMT data set. Just as for ISOT and Kaggle, LinearSVC also performed slightly better than XGBoost on a small training set. When its size exceeded 50 samples, XGBoost gained a large advantage.

Models trained on LIAR exhibited a still different behavior. The increase of the training set size had little impact on classification accuracy. Moreover, for both types of representations, the XGBoost model provided better performance.

The obtained results clearly show that the investigated fake news corpora differ significantly. Only Kaggle and ISOT are similar to each other. It also seems that FakeNewsNet has a lot in common with these data sets, but its size is two orders of magnitude lower. Most likely this is the reason why models trained on this corpus did not achieve high accuracy.

Moreover, for really small data sets, ranging from 5 to 50 samples, the representation by linguistic and psycholinguistic features provided better performance. It indicates that such representations are more universal. The advantage of linguistic representation over USE embeddings on small data sets might result from differences in their dimensionalities: 271 and 512, respectively. Furthermore, in the case of AMT, models trained on USE embeddings proved ultimately unsuccessful.

4.5 Cross-domain results

We also tested cross-domain performance: the models were trained on one data set and tested on other data sets. The results of 10-fold cross validation are presented in Table 7, where we can observe a substantial performance decrease of the classifiers within this test setting.

^kIn order to explore possible overfitting, we also examined the accuracy of AMT-based classifiers on their corresponding train data (unrelated to Figure 5). For a small number of samples in the training set (up to 10), the accuracy of LinearSVC on the training set is close to 1 and systematically decreases with the expansion of this data set, reaching a value of 0.8 (for 350 samples). The XGBoost classifier, however, reaches an accuracy of approx. 1 for the training set size of 10 samples and remains at the same level for a much larger training data set (350 samples). It seems that the addition of more training samples did not prevent overfitting and samples close to the ones that appear in the training set are given the same label. But, as the t-SNE visualization shows, if two samples are close to each other they often belong to different categories.

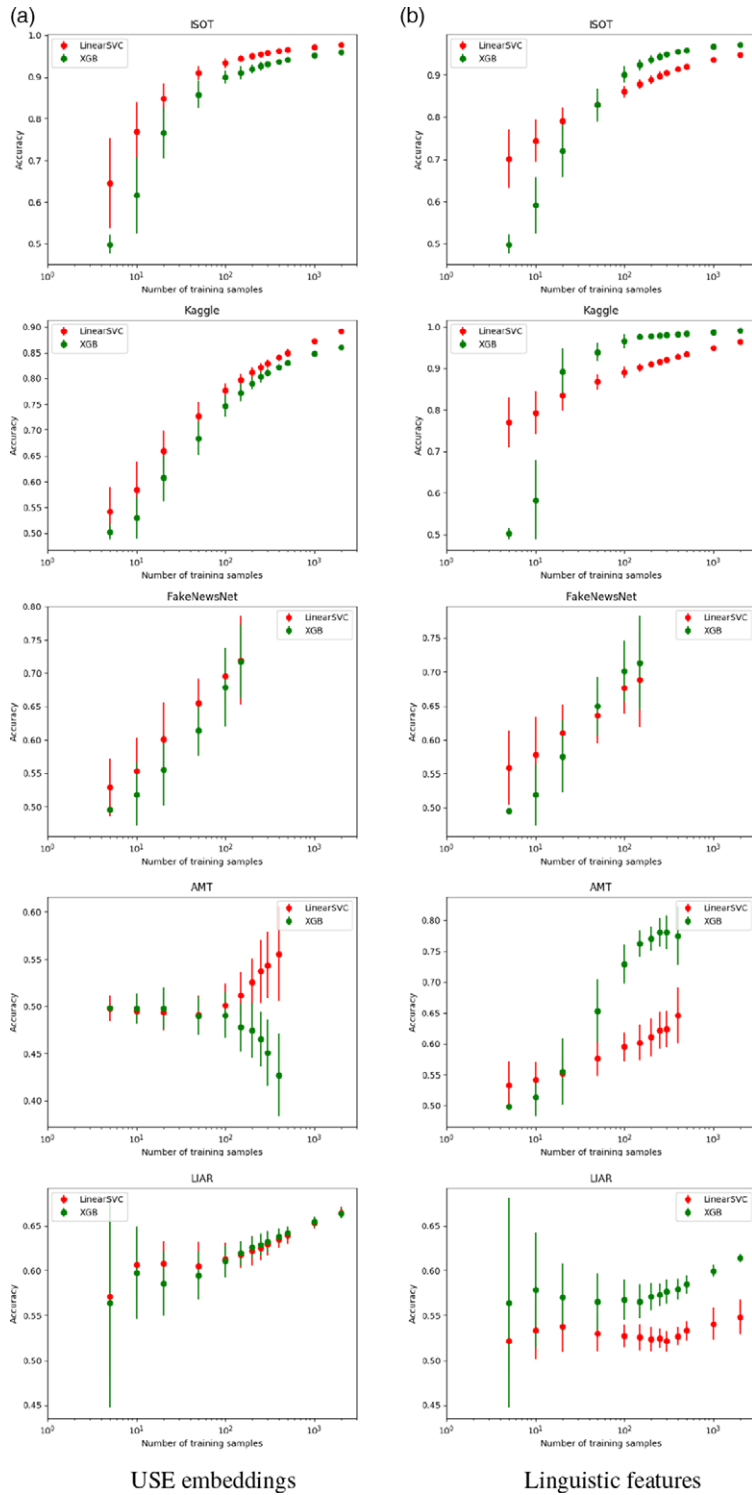


Figure 5. The dependence of accuracy of fake news detection on the size of training set.

Table 7. Cross-domain accuracy. Averages are calculated from cross-data set results (not in-domain test sets). Nonsignificant pairwise comparisons are marked with superscript letters (Wilcoxon test), nonsignificant group comparisons are greyed out (Cochran Q test)

Model	Train set	Tested on						Avg.
		Test set	Kaggle	ISOT	AMT	LIAR	FNN	
LinearSVC on BOW	Kaggle	0.97	–	0.76	0.61	0.62	0.64	0.66
	ISOT	0.99	0.63	–	0.57 ^b	0.63	0.54 ^c	0.59
	AMT	0.40	0.61 ^a	0.78	–	0.49	0.53 ^c	0.60
	LIAR	0.67	0.60 ^a	0.54	0.53	–	0.60	0.57
	FNN	0.75	0.67	0.63	0.57 ^b	0.53	–	0.69
							overall	0.60
XGBoost on linguistic features	Kaggle	0.99	–	0.51 ^a	0.47 ^b	0.63	0.52	0.53
	ISOT	0.98	0.66	–	0.49 ^c	0.62	0.63	0.60
	AMT	0.80	0.42	0.38	–	0.52 ^c	0.57	0.47
	LIAR	0.63	0.73	0.54 ^a	0.47 ^{b,c}	–	0.50	0.56
	FNN	0.70	0.62	0.66	0.57	0.50 ^c	–	0.59
							overall	0.55
LinearSVC on USE	Kaggle	0.93	–	0.69	0.50	0.59	0.64	0.61
	ISOT	0.99	0.55	–	0.54 ^b	0.61	0.50	0.55
	AMT	0.57	0.43	0.62 ^a	–	0.47 ^c	0.57	0.52
	LIAR	0.68	0.52	0.53	0.50 ^b	–	0.50	0.51
	FNN	0.72	0.65	0.60 ^a	0.52	0.48 ^c	–	0.56
							overall	0.55
Dense on USE	Kaggle	0.92	–	0.71	0.51	0.53	0.68	0.61
	ISOT	0.99	0.56	–	0.57	0.59	0.53	0.56
	AMT	0.51	0.47	0.65 ^a	–	0.48 ^c	0.60	0.55
	LIAR	0.68	0.52	0.52	0.53 ^b	–	0.52	0.52
	FNN	0.74	0.65	0.61 ^a	0.52 ^b	0.47 ^c	–	0.56
							overall	0.56
Bi-LSTM on GloVe	Kaggle	0.91	–	0.73	0.56	0.63 ^{e,f}	0.54 ^{i,j}	0.62
	ISOT	0.99	0.70	–	0.59	0.63 ^{e,g}	0.56 ^{i,k}	0.62
	AMT	0.56	0.52 ^a	0.52 ^{b,c}	–	0.52 ^h	0.55 ^{i,k}	0.53
	LIAR	0.63	0.49	0.52 ^{b,d}	0.50	–	0.50	0.50
	FNN	0.73	0.53 ^a	0.53 ^{c,d}	0.51	0.58 ^{f,g,h}	–	0.54
							overall	0.56

The data set-specific accuracy of classifiers trained on Kaggle and ISOT exceeded 0.9. However, when the same classifiers were tested on data sets different from the one used for training, the accuracy decreased to even slightly below 0.5 in some cases. This shows that those models are good in recognizing fake news within one data set, but not fake news in general, that is independently from training data. The most universal method which achieved the highest accuracy averaged over all train–test pairs turned out to be LinearSVC on BOW. In general, the highest scores in cross-domain setting were obtained for the ISOT-Kaggle pair due to the similarity of both data sets. The lowest accuracy of models trained on Kaggle and ISOT was achieved by XGBoost with linguistic features tested on the AMT data set. While Kaggle and ISOT data sets are similar in terms of size and structure, they significantly differ from the AMT data set. However, the best method was LinearSVC on BOW. It does not mean that the topics were the same. Most likely, similar stylistic means were used in fake news across various data sets, which is recognized by this classifier.

Models trained on AMT showed the most unpredictable results. LinearSVC with BOW features, which obtained very low accuracy in data set-specific settings, turned out to perform very well on ISOT data. This result is surprising, with 0.4 accuracy on its test data and 0.78 on the ISOT test set. We examined these results, looking for the words that have the greatest impact on the predicted label. The AMT’s most significant n-gram features were neutral words such as “their”, “be”, “this”, “they”, and “are”. This was due to the way the data set was created. For each true text, there is a corresponding false one on the same topic, as we explained in Section 3. These influential words are not as abundant in the AMT test set, but they are consistent with the fake news style in other data sets, which explains why the results vary so much.

As far as the linguistic features are concerned, the classifier trained and tested on AMT obtained good results (0.80). At the same time, when tested on Kaggle and ISOT, their accuracy was 0.42 and 0.38, respectively, which were the lowest cross-domain results. Also, we can observe that some classifiers trained on different data sets managed to perform better on AMT than classifiers trained on AMT. For instance, LinearSVC using bag-of-words achieved an accuracy of 0.57 on AMT when trained on FakeNewsNet, and 0.61 when trained on Kaggle, which was higher by 0.21 than its results for data set-specific settings. This shows that the small size of the AMT data set hinders its effective training.

On the LIAR data set, the highest cross-domain accuracy was 0.63, achieved by several classifiers.

In general, the decline in the quality of the model predictions in this scenario compared to the in-domain setting turned out to be very large. This phenomenon is at least partly due to the heterogeneous content of the data sets, containing texts of different nature, topics, and structure. Different ways of defining and creating false and true texts, as well as the time of their writing, may also explain this decline in quality.

5. Leave-one-out settings

Another interesting variation of the domain adaptation setting we decided to investigate is the leave-one-out scenario. In this strategy, a classifier is trained on all but one selected data set, which is further used for validation of the system. Such an approach might help smooth out some of the differences between the data sets as well as provide more training data. As the number of samples in the investigated data sets differs substantially, we decided to consider two approaches:

- (1) **All:** all samples from data sets are used for training. Therefore, for a different selection of the train and validation data sets, their sizes differ.

Table 8. Leave-one-out accuracy

Model	Tested on	Train set preparation strategy			
		All	Balanced	Average	Best
LinearSVC on BOW	Kaggle	0.66	0.72	0.66	0.76
	ISOT	0.63	0.65	0.59	0.63
	AMT	0.61	0.64	0.62	0.80
	LIAR	0.61	0.59	0.58	0.62
	FNN	0.57	0.63	0.61	0.68
XGBoost on linguistic features	Kaggle	0.60	0.42	0.54	0.63
	ISOT	0.53	0.62	0.60	0.67
	AMT	0.42	0.51	0.48	0.57
	LIAR	0.63	0.63	0.57	0.79
	FNN	0.63	0.69	0.58	0.64
LinearSVC on USE	Kaggle	0.58	0.62	0.60	0.69
	ISOT	0.68	0.73	0.55	0.61
	AMT	0.53	0.52	0.52	0.62
	LIAR	0.61	0.50	0.52	0.53
	FNN	0.62	0.61	0.57	0.66

- (2) **Balanced:** the training corpus is composed of the same number of samples from each data set. As the smallest data set—FakeNewsNet—contains only 182 samples, such a number of randomly selected samples from each data set was used to create the training data. Therefore it always contains $4 \times 182 = 728$ samples;

The results of this approach are presented in Table 8. Moreover, this table contains two columns—Average and Best—which refer to the results presented in Table 7: the best and averaged results obtained for corresponding models and test sets in the default cross-domain setting. In general, the results of the balanced setting are better than for models trained on all samples from data sets selected for training. It is worth noting that the amount of training data is relatively small when using this configuration. Typically the accuracies of models trained on a balanced mixture of data sets are higher than the average accuracies obtained in cross-domain settings reported in Table 7. The leave-one-out settings resulted in the highest accuracy in only two cases: XGBoost classifier using linguistic features tested on FNN, and LinearSVC classifier utilizing USE embeddings validated on ISOT.

6. Fact-checking

A completely different approach to detecting fake news is automated fact-checking. A discussion of possible approaches is presented in Section 2.1. Here, we discuss the method we applied to fact-check the data sets described in Section 3. The system consists of a knowledge base that is

Table 9. Fact-checking results by Team Domlin system (Stammbach and Neumann 2019)

Dataset	Texts	Texts w/o NEI-only	Accuracy w/o NEI-only
Kaggle	2000	749	0.53
ISOT	2000	283	0.45
FakeNewsNet	182	51	0.41
AMT	720	54	0.63
LIAR	2000	16	0.64

considered to be true, an information retrieval module to look up relevant articles, and a neural network trained on the FEVER data to perform inference (Thorne *et al.* 2018).

The selected system is the one presented by Team Domlin (Stammbach and Neumann 2019) during the second edition of the FEVER competition (Thorne *et al.* 2019). The system was built using the FEVER data set (Thorne *et al.* 2018) for the FEVER competition, which requires for a given claim to either find evidence to support/reject that claim or, if insufficient evidence is found, classify it as ‘Not Enough Info’ (NEI). The database for evidence searching is Wikipedia. The Domlin system uses BERT representations (Devlin *et al.* 2019) to select from Wikipedia sentences that are most relevant to the given claim, and then another BERT model to decide whether gathered evidence supports or refutes the claim. This model achieved one of the best results in the FEVER competition and is able to find evidence for claims outside of the FEVER data set (is able to generalize), although the usage of Wikipedia as a source of ground truth data has severe limitations.

Fact-checking systems often process sentence-level information. The data sets in Section 3 (except LIAR) typically contain many sentences per text and the veracity is attributed to the whole text, not individual sentences. In order to compute text-level accuracy of fact-checking, we implemented the following principle: if a text contains at least one sentence labeled as ‘Refuted’ (presumably a false sentence), classify the whole text as false. If a text contains at least one sentence labeled as ‘Supported’ (presumably a true sentence) and there is no sentence labeled as ‘Refuted’, classify the text as true. Texts with contradictory labels were classified as fake due to the fact that fake pieces of news are often mixtures of true and fake information.

Due to long processing times, we had to limit the size of the large data sets (ISOT, Kaggle, and LIAR). Each was randomly sampled for 1000 true and 1000 fake texts. Smaller data sets (FakeNewsNet and AMT) were used in full. Table 9 reports the numbers of texts with at least one sentence labeled other than ‘Not Enough Info’ (NEI) by the Team Domlin system (Texts w/o NEI-only), as well as the accuracy achieved on those texts (Accuracy w/o NEI-only). In other words, for computing accuracy we removed all texts where all sentences obtained NEI labels.

The results achieved on these data sets are far from satisfactory. The biggest problem is that most of the sentences obtained ‘Not Enough Info’ (NEI) labels by the Team Domlin system. The number of sentences with labels other than NEI, suitable for veracity inference, was between 1% and 2% depending on the data set. This translates to a low number of texts with at least one non-NEI sentence, far too low to consider this form of veracity verification satisfactory for real-world application. One can hypothesize that the biggest obstacle is that the tested fact-checking system is limited to the content of Wikipedia, which is neither sufficient nor suitable for real claims on the internet.

7. Feature selection

Feature selection is often used to design more accurate models and seek explainable architectures. It also can serve as a regularization technique since it decreases model dimensionality. In this section, we describe feature selection algorithms that were applied to improve cross-domain performance. First, we analyzed a subset of features that were important across all data sets, and next we designed a classifier based on the intersection of the top features.

We compared two algorithms of feature selection: Mutual Information (MI) (Kraskov, Stögbauer, and Grassberger 2004) and Minimum Redundancy Maximum Relevance (MRMR) (Peng, Long, and Ding 2005). The former is a univariate filter method: each feature is evaluated individually according to specific criteria, and no interactions between features are considered. Then, the most relevant features are selected, which is called a *maximum-dependency* or *maximum-relevance* scheme. Correlation or mutual information is typically used as a measure of a feature's importance, based on which features are ranked. We decided to use Mutual Information, which measures the dependency between variables (in our case between each feature and target) based on an entropy estimation of the distance to the k nearest neighbors. We used the implementation provided by the *scikit-learn* library (Pedregosa *et al.* 2011).

However, certain studies (Peng *et al.* 2005) have shown that combining individually good features does not always result in good classification performance. MRMR is a heuristic algorithm that aims at finding a subset of features that is close to the optimal one. Contrary to MI, it also takes into account the correlations between variables. Features are selected in such a way that they correlate very strongly with the target (maximum-relevance), but are mutually as dissimilar to each other as possible (minimum-redundancy condition). The introduction of the latter condition often enriches the information provided by MI.

In the first experiment, we selected n top features for each combination of feature space, selection method, and data set. Next, we checked which features were common among all data sets. We tested different values of n on both linguistic and bag-of-words features. The whole process along with the achieved results is presented in Subsection 7.1.

The second experiment utilized the cross-domain feature selection mechanism described above to enhance the performance of the model. The model was trained on one data set and tested on another, but only features from the intersection of the top features from the remaining three data sets were used for training. For the rationale behind such an approach as well as a detailed description of the procedure, see Subsection 7.2.

7.1 Intersection of top features

During the search for features that are important for fake news detection across all data sets, we investigated the intersection of top n ($n = 20, 50, 100, 150, 200$) linguistic features from each data set. As Table 10 shows, the Mutual Information method—compared with MRMR—resulted in a considerably lower number of important common features across all five data sets. The intersection of the top 20 features turned out to be empty and there was only one feature in the top-50 intersection: TIME. This category comes from the General Inquirer dictionary and describes words indicating time consciousness. Selected features are shown in Table 11.

When the MRMR method was used, two POS features turned out to be present in the intersection of the top 20 features: the percentage of nouns and adjectives in the text. The third feature found (`complex_words`) was the ratio of complex words to all words in the document. This is one of the indicators used to measure text readability.

Figure 6 shows the frequencies of three relevant parts of speech: nouns, adjectives, and numerals. The importance of the percentage of nouns in fake news classification is in line with the research by Horne and Adali (2017). Their study revealed that the number of nouns is lower in fake news than in real news, and close to that found in satire. Figure 6a illustrates that all data

Table 10. The number of top *n* features common for all data sets (5)

Number of top features	Linguistic features		BOW features	
	Mutual information	MRMR	Mutual information	MRMR
20	0	3	10	0
50	1	30	31	0
100	6	68	57	0
150	26	116	74	0
200	94	157	96	0
300	–	–	131	0
500	–	–	193	0

Table 11. Intersection of top *n* linguistic features

Mutual information	MRMR
20	20
50	50
100	100
20	20
50	50
100	100

sets except LIAR showed this trait. The LIAR data set consists of claims collected from transcripts, speeches, news stories, press releases, and campaign brochures¹ by PolitiFact journalists, which makes the LIAR texts different in length and style from the other data sets.

When it comes to the average percentage of adjectives in a sentence (Figure 6b), we can observe differences in feature distribution between classes for Kaggle and ISOT data sets. However, in the Kaggle data set fake news had more adjectives than in real news, while in the ISOT data set it was the opposite—real news had on average more adjectives than fake news. It is therefore hard to draw any conclusions about the role of adjectives in distinguishing fake and real news based on these facts alone. One possible continuation of this research may be to perform expert analysis on journalistic style traits present in different data sets, which may explain some discrepancies in feature distribution. However, our aim is to explore the possibility of a universal classification

¹<https://www.politifact.com/truth-o-meter/article/2018/feb/12/principles-truth-o-meter-politifact-methodology-i/>

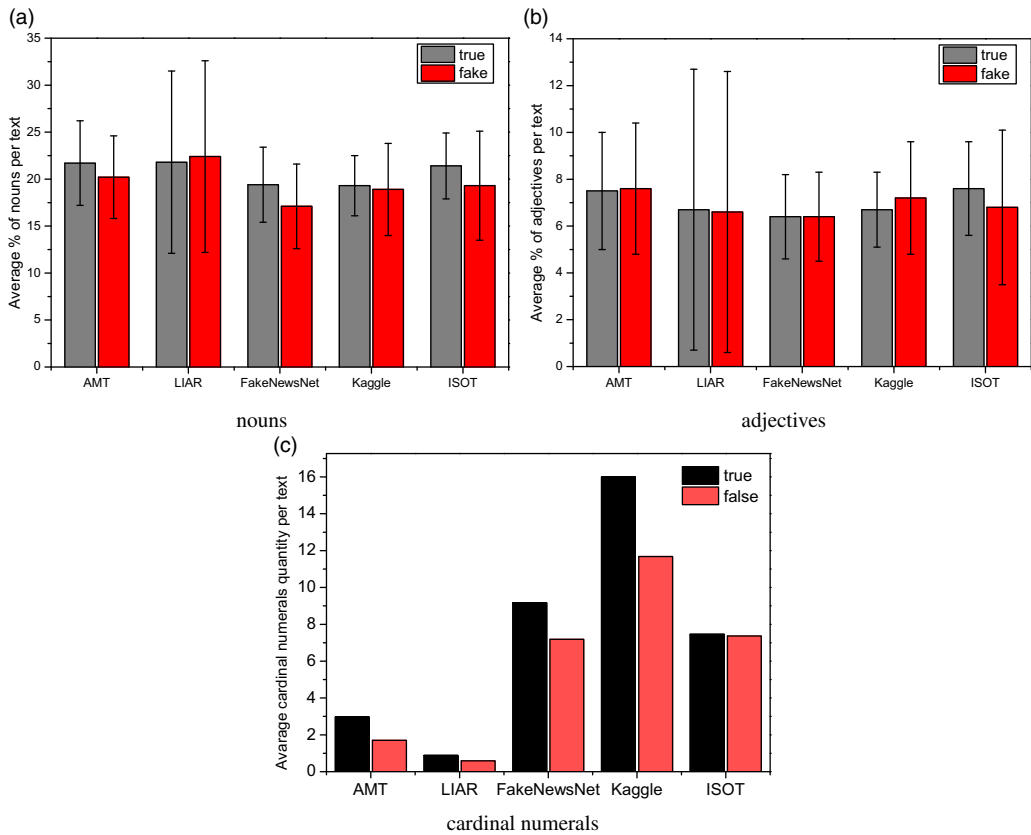


Figure 6. Part-of-speech statistics.

model in the area of fake news detection, so we put aside differences in news writing styles to be examined in more depth in future research.

Aside from the number of complex words, other readability indicators also turned out to be important when it comes to fake news classification. The intersection of the top 50 features from each data set (30 features) contained 23% readability-related features, whereas readability indicators made up only 7% of all features. The well-known readability measures are shown in Table 11 and consist of the Gunning–Fog index (Gunning 1969), the number of long words, and the type-token ratio (lexical diversity), among others.

We can hypothesize that to fulfill their role of misinformation, fake messages need to be easily understood, whereas real news seeks to express the complexity of the reported events and thus may require more sophisticated language. The importance of readability in fake news detection is also confirmed by Pérez-Rosas *et al.* (2018) and Horne and Adali (2017). What is more, it also proved to be true for different languages (Santos *et al.* 2020). However, the hypothesis about the source of differences in text readability between fake and real news should be further investigated via sociolinguistic studies.

Analogically, we investigated the intersection of bag-of-words features. We checked how many of the top n features ($n=20, 50, 100, 150, 200, 300, 500$) were common for all data sets. The results are presented in Table 10. As we can see, the two feature selection methods yielded distinctly different results. Mutual Information provided intersections ranging between 39 and 57% of the top features. For example, considering the top 20 features selected for each data set, half of them

(10) were common for all data sets. This may be surprising when we consider that the BOW feature space is very vast, containing thousands of n-grams. For MRMR, we decided to take into consideration only the 5000 most frequent features in a given corpus to prevent the algorithm from selecting the rare ones. Nevertheless, the set of common features selected with the MRMR algorithm was empty for every n value.

In Figure 6c, we can see the general tendency of fake news to contain fewer cardinal numerals, which holds for every data set. In Horne and Adali (2017), the authors observed that fake news contains fewer technical and analytical words. Our findings are in line with their study showing that fake news is less likely to present concrete numerical data, which are rather common in technical or analytical writing.

The results of the MI feature selection on BOW data are consistent with the observation made by investigating linguistic features (see Table 11), where the percentage of pronouns, prepositions, and conjunctions per text was selected as an important linguistic feature that was common for every data set.

Furthermore, we checked the distribution of these features within each class and observed the tendency for fake news to have on average more pronouns than real news, which was especially visible in the AMT, ISOT, and FNN data sets. A more in-depth analysis of pronoun distributions in fake news versus real news conducted by Yang *et al.* (2018) showed fewer first- and second-person pronouns in fake news, but a higher number of third-person pronouns. Considering the average number of prepositions, we may observe that they were dominant within the real news class in LIAR, Kaggle, and ISOT data sets, and only AMT showed the opposite pattern. According to Santos *et al.* (2020), prepositions are indicators of text cohesion, which is further studied in the paper for the Brazilian Portuguese language. Our result can therefore be a premise to conduct such research for the English language as well.

The purpose of this experiment was to investigate if there is a way to create a classifier that will show similar accuracy independent of the origin of the test set based on five different data sources. We used feature selection methods to find out which features are good indicators of fake or real news across all five data sets. Further, we compared selected feature distributions between classes within the studied data sets. In some cases (nouns, pronouns, cardinal numbers, prepositions), we found traits common for most of the data sets, whereas in other cases (e.g. adjectives) features played an important role for most data sets but the class they indicated differed between data sets. One of the plausible continuations of this research is to investigate the differences between journalism styles and compare them to the style of fake news or possibly cluster fake news based on linguistic features and try to uncover some differences in the style of fake news itself.

7.2 Intersection-based classifier

In this subsection, we present the results of classifiers based on the intersection of the top features. We trained a model on one data set using selected features and tested it on another. The procedure of feature selection was similar to the one presented in the previous subsection; however, only three data sets were used to create the top feature intersection instead of five. The test set was not supposed to take part in the creation of the model, so it could not be a part of the process of feature selection. Also, the training set was excluded since selecting top features common for several data sets is meant to reduce the bias towards training data and provide more universal results.

Moreover, a lower number of data sets in the process of feature selection resulted in a larger number of selected features. For example, the five-data set intersection of the top 50 linguistic features selected with the MI algorithm had only one feature (see Table 10), whereas for three data sets the number of features varied between 1 and 7 (depending on the selected data sets). Similarly, when the same settings were considered for bag-of-words features, the intersection of all five data sets contained 30 items (see Table 10), while the number of common features for three data sets varied between 31 and 42.

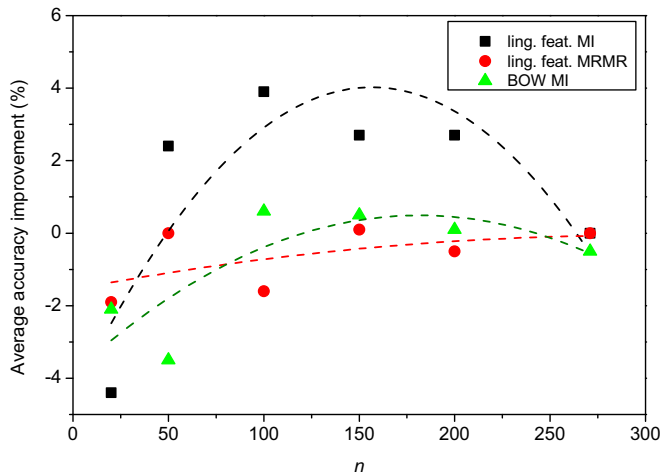


Figure 7. Average cross-domain accuracy gain for various numbers of top features (n).

For the intersection-based classifiers, we chose the same algorithms as those described in Subsection 4.2: XGBoost on linguistic features and LinearSVC on BOW. We tested them in a cross-domain manner, as in Subsection 4.5, but using only selected features. For each train-test pair, a set of features was selected based on the intersection of the top features of the three remaining data sets. For linguistic classifiers, we considered two methods of selecting top features: Mutual Information and Minimum Redundancy Maximum Relevance. With the bag-of-words feature space the MRMR method selected very different features for each data set, so the intersection was empty (even for only three data sets). Therefore, we could not create an intersection-based classifier for BOW using MRMR and we used only the Mutual Information algorithm.

We adopted average gain as the measure of the overall performance of a given feature selection setting. To calculate the overall performance, we first subtracted full-feature model cross-domain results (Table 7) from the results of the intersection-based classifiers for each train-test pair. Then, we added up the individual gains and divided the result by the total number of pairs (20) to obtain the average gain.

Figure 7 shows average gains for varying number of top features (n). We can observe that for the MRMR method, none of the investigated values of the n parameter provided any improvement in the results. As for the Mutual Information method, $n=100$ provided the best results for both linguistic and BOW features. On average, the accuracy improved by 4% and 1%, respectively.

Table 12 presents the cross-domain results of classifiers trained on the intersection of the top 100 features selected with the Mutual Information method. In the last column, we present the average gain for all pairs with a given data set used for training, while the last row presents the average gain for all pairs when the data set was used for testing.

Linguistic classifiers showed an overall greater improvement (4% averaged over all train-test pairs) than the ones based on BOW (average gain 1%). The accuracy improved for almost every training set, with the most significant gain for Kaggle (6%) and AMT (11%). This proves that a carefully designed feature space, which takes into account feature importance computed for several different data sets, can improve generalization of a model trained on a particular data set. Although the BOW classifier had a lower average gain than the linguistic one, it showed higher accuracy, especially when tested on Kaggle, ISOT, and AMT. As the BOW classifier performed better from the beginning, there was less room for improvement. This may explain why this method was less effective in shifting the cross-domain performance.

Table 12. Accuracy of classifiers based on Mutual Information, intersection of the top 100 linguistic features

Features	Train set	Kaggle	ISOT	AMT	LIAR	FNN	Avg. gain (%)
Ling	Kaggle	–	0.53	0.55	0.63	0.66	6
	ISOT	0.63	–	0.53	0.62	0.64	1
	AMT	0.62	0.56	–	0.60	0.57	11
	LIAR	0.48	0.69	0.50	–	0.57	–1
	FakeNewsNet	0.65	0.68	0.59	0.47	–	1
	Avg. gain (%)	–3	10	4	1	5	4
	BOW	Kaggle	–	0.75	0.64	0.62	0.66
ISOT		0.64	–	0.63	0.56	0.60	1
AMT		0.67	0.77	–	0.52	0.66	4
LIAR		0.55	0.46	0.52	–	0.52	–6
FakeNewsNet		0.70	0.73	0.60	0.56	–	3
Avg. gain (%)		0	–2	3	0	2	1

8. Cross-domain performance enhancements

We also investigated more complex methods designed to compensate for different distributions of train and test data. There is a whole group of domain adaptation techniques specifically addressing this problem which is closely associated with machine learning and transfer learning. The goal of such algorithms is to train a model with labeled data in a source domain that will perform well on different (but related) target data.

In supervised learning (without domain adaptation), we usually assume that both training and test set examples were drawn from an identical or very similar distribution. In the domain adaptation scenario, we considered two different (but related) distributions: one for the source data and one for the target data. The domain adaptation task then consisted of a transfer of knowledge from the source domain to the target domain, minimizing errors on the target domain.

In the machine learning community, issues similar to that of domain adaptation have been studied under the term “dataset shift” (Quionero-Candela *et al.* 2009). This well-known problem of predictive modeling occurs when the joint distribution of inputs and outputs differs between training and test stages. A data set shift may have multiple causes, ranging from a bias introduced by the experimental design to the irreproducibility of the testing conditions at the training stage. In our case, the data set shift was related to different data sources and time periods of the data sets as well as varying topics and structure of the texts. Ensuring domain adaptation (preventing a data set shift) often involves matching distributions so that the training (source) data distribution more closely matches that of the test (target) data. As our aim was to improve the cross-domain performance, we used one data set as the source domain (train set) and another as the target domain (test set). The results of our experiments are described in Sections 8 and 9. We used a 10-fold cross-validation and performed statistical significance tests.

The selected machine learning approaches that achieved this goal are grouped into the following types and described in-depth below:

- **Instance re-weighting** – The goal of instance reweighting (IRW) is to use the source data for training while optimizing performance on the target data (Jiang and Zhai 2007).

Table 13. Domain adaptation accuracy on models with linguistic features. Nonsignificant pairwise comparisons are marked with superscript letters (Wilcoxon test), nonsignificant group comparisons are greyed out (Cochran Q test)

Method	Features	Train set	Kaggle	ISOT	AMT	LIAR	FNN	Avg. gain (%)
GFK (<i>dim=10</i>)	Ling	Kaggle	–	0.57	0.52	0.61	0.54 ^d	3
		ISOT	0.60 ^a	–	0.56	0.53	0.53 ^d	–4
		AMT	0.53 ^{a,b}	0.65 ^c	–	0.49	0.68	11
		LIAR	0.52 ^b	0.52	0.50	–	0.50	–4
		FNN	0.66	0.65 ^c	0.58	0.55	–	2
		Avg. gain (%)	–3	8	4	–2	1	2
SA (<i>loss=logistic</i> , <i>num comp=4</i>)	Ling	Kaggle	–	0.53 ^d	0.52	0.52	0.63	2
		ISOT	0.71 ^{a,b}	–	0.50 ^e	0.52 ^f	0.49	–4
		AMT	0.79 ^{a,c}	0.67 ^e	–	0.52 ^f	0.61	17
		LIAR	0.66	0.52 ^d	0.50 ^e	–	0.51	0
		FNN	0.70 ^{b,c}	0.64 ^e	0.54	0.50	–	1
		Avg. gain (%)	11	8	2	–5	1	3
IRW (<i>loss=xgboost</i> , <i>iwe=lr</i> , <i>l2=1.0</i>)	Ling	Kaggle	–	0.51	0.48	0.63	0.49 ^d	0
		ISOT	0.65	–	0.54 ^a	0.63	0.57 ^e	0
		AMT	0.39	0.41	–	0.50 ^c	0.55 ^{e,f}	–1
		LIAR	0.72	0.48	0.52 ^b	–	0.52 ^{d,f}	1
		FNN	0.62	0.64	0.54 ^{a,b}	0.47 ^c	–	–2
		Avg. gain (%)	–1	0	2	–1	–2	0

Such optimizations can be based on assigning higher weights to training instances that are close to the target instances; the assumption being that they are of greater importance to the classification of the target data. Re-weighting each source instance in such a way approximates risk minimization under the target distribution. We applied this approach using an XGBoost classifier on linguistic features and LinearSVC on USE embeddings. We used the implementation from libTLDA library^m and selected the optimal loss functions.

- **Common representation space** – Another approach to domain adaptation is based on finding a domain-invariant feature space. This can be achieved by creating subspaces for both domains and aligning the source one with the target one. We tested two methods belonging to this class:
 - Subspace alignment (SA) (Fernando *et al.* 2013): one of the most straightforward methods of finding common subspaces. For each domain, the first d principal components, C_{src} and C_{trg} , are computed. A linear transformation matrix that aligns source components to target components is defined as $M = C_{src}^T C_{trg}$. The adaptive classifier initially projects data from each domain to their corresponding components and is trained

^m<https://pypi.org/project/libtlda/>.

Table 14. Domain adaptation accuracy on models with USE embeddings. Nonsignificant pairwise comparisons are marked with superscript letters (Wilcoxon test), nonsignificant group comparisons are greyed out (Cochran Q test)

Method	Features	Training set						Average gain (%)
		Kaggle	ISOT	AMT	LIAR	FNN		
GFK <i>(dim=20)</i>	USE	Kaggle	–	0.68 ^a	0.52	0.49	0.67	–1
		ISOT	0.59	–	0.54	0.57	0.52	0
		AMT	0.46	0.71 ^a	–	0.47	0.60	4
		LIAR	0.49	0.52	0.50 ^b	–	0.50	–1
		FNN	0.66	0.60	0.51 ^b	0.53	–	1
		Avg. gain (%)	1	2	0	–2	2	1
SA <i>(loss=logistic, num comp=11)</i>	USE	Kaggle	–	0.72 ^a	0.50	0.43	0.62	–4
		ISOT	0.65	–	0.51	0.49 ^c	0.59	1
		AMT	0.50	0.69 ^a	–	0.52 ^{c,d}	0.47 ^e	2
		LIAR	0.54	0.53 ^b	0.50	–	0.45 ^e	–1
		FNN	0.62	0.53 ^b	0.49	0.48 ^d	–	–3
		Avg. gain (%)	4	1	–2	–6	f–2	–1

on the projected source data transformed using matrix M . Again, we used the implementation from libTLDA library and tested various loss functions and a number of components.

- Geodesic flow kernel (GFK) (Gong *et al.* 2012): a more sophisticated method based on the assumption that a manifold of transformations exists between the source and the target domains. This path is defined by the projection matrices, $\Phi(t)$, where $t \in [0, 1]$. At $t = 0$ the projection consists purely of the source components C_{src} , while at $t = 1$ it consists exclusively of the target components C_{trg} . Geodesic flow kernel incorporates the entire path of these transformations forming a kernel:

$$G(x_i, x_j) = \int_0^1 x_i \Phi(t) \Phi(t)^T x_j^T dt,$$

where x_i and x_j are two feature vectors. The resulting kernel can be used to construct any kernelized classifier such as a support vector machine.

8.1 Results of domain adaptation methods

The results of the above described domain adaptation machine learning techniques are presented in Tables 13 and 14. They were applied to two feature spaces: linguistic features and embeddings obtained from the Universal Sentence Encoder. Figure 8 visualizes the change in classification accuracy for each train–test pair upon application of domain adaptation methods. Positive values are marked in red, while negative values are marked in blue. As a reference, XGBoost results were used for classifiers trained on linguistic features. LinearSVC was used for models utilizing USE embeddings.

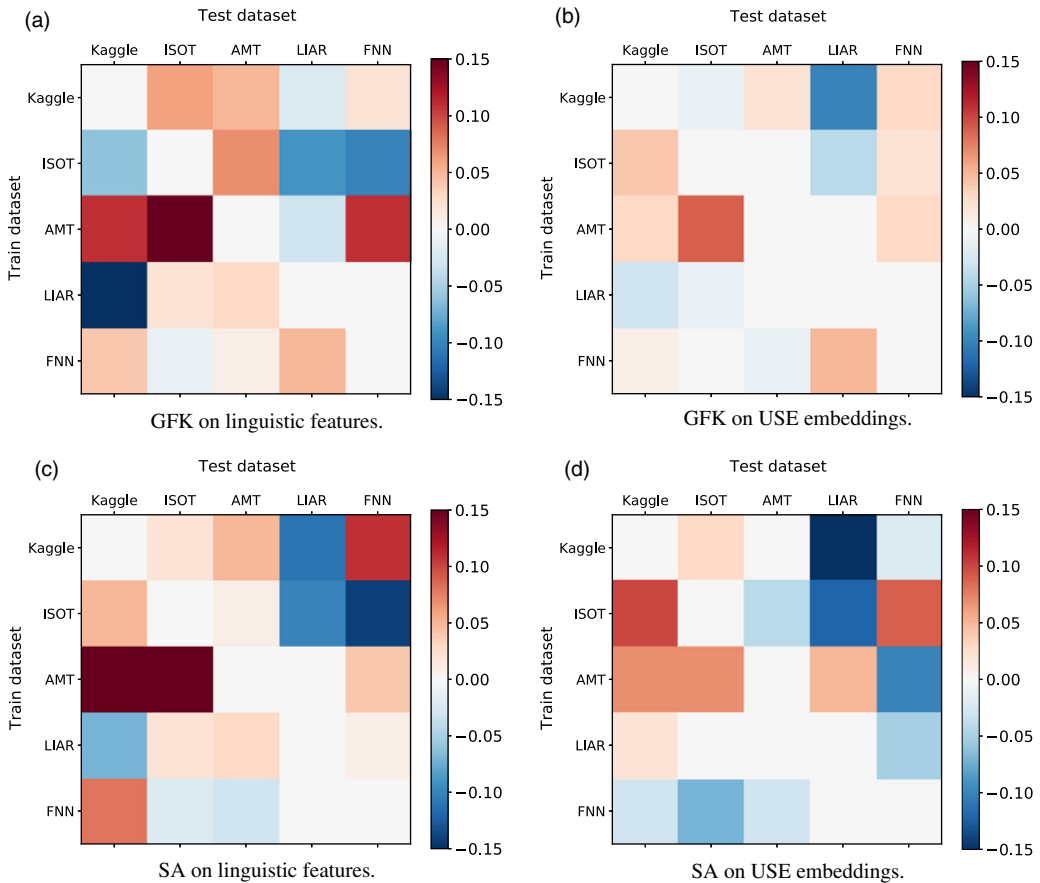


Figure 8. Improvement in accuracy achieved by different domain adaptation methods applied to two feature spaces: USE embeddings and linguistic features.

Unfortunately, there seems to be no universal method that would guarantee an improvement for all configurations of the train–test pairs. While there was a significant increase in accuracy for some pairs, others showed a decrease in performance. In general, the domain adaptation methods provided a higher improvement for linguistic classifiers than for the ones based on USE embeddings. The highest overall improvement (averaged over all train–test pairs), equal to 3%, was achieved by subspace alignment applied to a classifier based on linguistic features.

9. Deep domain adaptation

We also explored selected deep domain adaptation methods on the task of fake news recognition. Neural networks are very powerful tools due to their ability to learn and recognize patterns. Thanks to their high capacity, they have gained immense popularity and achieved state-of-the-art results in numerous NLP tasks such as machine translation, named entity recognition, language modeling, text classification, etc. However, the performance of these deep learning models can also suffer from domain shift. Therefore, much research has been devoted to the development of methods aiming to adapt neural networks trained on a large number of labeled source data to a target domain for which no labels are available. Thanks to such an approach, the labeling of data from the target domain, which often requires considerable resources, can be avoided. Most of

Table 15. Domain adaptation cross-domain accuracy for deep learning methods applied to the model using USE embeddings. Nonsignificant pairwise comparisons are marked with superscript letters (Wilcoxon test), nonsignificant group comparisons are greyed out (Cochran Q test)

Method	Features	Train set	Kaggle	ISOT	AMT	LIAR	FNN	Avg. gain (%)
ADR	USE	Kaggle	–	0.74	0.50 ^c	0.63	0.67	3
		ISOT	0.63 ^a	–	0.54	0.63	0.52	2
		AMT	0.51	0.53 ^b	–	0.49 ^d	0.54	–3
		LIAR	0.49	0.52 ^b	0.50 ^c	–	0.50	–2
		FNN	0.62 ^a	0.59	0.52	0.49 ^d	–	–1
		Avg. gain (%)	1	–3	–2	4	–2	0
VADA	USE	Kaggle	–	0.72	0.52 ^c	0.49 ^e	0.70	0
		ISOT	0.60	–	0.55	0.54 ^f	0.53 ^h	2
		AMT	0.50 ^a	0.63 ^b	–	0.49 ^{e,f,g}	0.52 ^{h,i}	–2
		LIAR	0.51 ^a	0.53	0.52 ^{c,d}	–	0.51 ⁱ	0
		FNN	0.65	0.60 ^b	0.53 ^d	0.51 ^g	–	1
		Avg. gain (%)	2	0	0	2	–2	0
Coral	USE	Kaggle	–	0.71	0.50	0.53	0.67	0
		ISOT	0.52 ^{a,b}	–	0.56	0.57	0.58	0
		AMT	0.51 ^{a,c,d}	0.63 ^f	–	0.49 ^h	0.67	2
		LIAR	0.52 ^{b,c,e}	0.54	0.53 ^g	–	0.50	0
		FNN	0.51 ^{d,e}	0.60 ^f	0.53 ^g	0.45 ^h	–	–4
		Avg. gain (%)	–4	0	0	–1	2	0

the deep domain adaptation methods were originally designed and tested for computer vision tasks such as object detection and classification (digit, traffic signs, etc.). These methods aimed at adaptation of models that were trained on one type of images to perform well on other types. For instance, synthetic images and real pictures taken in different conditions (presence of background, different lighting, etc.) were involved. An extensive review of deep domain adaptation methods was conducted by Wilson and Cook (2018). Domain adaptation algorithms were also tested in the field of natural language processing. They were applied mainly for text classification (Liu, Qiu, and Huang 2017), including sentiment analysis, but also for relation extraction (Fu *et al.* 2017) and machine translation (Chu and Wang 2018).

We investigated the following deep domain adaptation techniques:

- Deep correlation alignment (CORAL, Sun and Saenko 2016): unsupervised domain adaptation method that aligns the second-order statistics of the source and target distributions. The nonlinear transformation is found using a deep neural network by introducing additional loss (CORAL loss), which measures the distance between covariances of the learned source and target features.
- Adversarial dropout regularization (ADR, Saito *et al.* 2018): extension of the concept of reducing the discrepancy between the source and target feature distributions by adversarial

Table 16. Domain adaptation accuracy for deep learning methods applied to the model using GloVe embeddings. Nonsignificant pairwise comparisons are marked with superscript letters (Wilcoxon test), nonsignificant group comparisons are greyed out (Cochran Q test)

Method	Features	Train set	Kaggle	ISOT	AMT	LIAR	FNN	Avg. gain (%)
ADR	GloVe	Kaggle	–	0.65	0.57	0.63 ^{d,e}	0.62	0
		ISOT	0.55 ^a	–	0.52 ^c	0.63 ^{d,f}	0.54	–6
		AMT	0.54 ^a	0.48	–	0.53 ^g	0.51 ^h	–1
		LIAR	0.49 ^b	0.52	0.50	–	0.50 ^h	0
		FNN	0.49 ^b	0.60	0.52 ^c	0.55 ^{e,f,g}	–	0
		Avg. gain (%)	–4	–1	–1	0	0	–1
VADA	GloVe	Kaggle	–	0.72	0.59	0.63 ^d	0.64	3
		ISOT	0.57	–	0.53	0.61 ^{d,e,f}	0.54	–6
		AMT	0.53	0.52 ^b	–	0.55 ^{e,g}	0.49 ^h	–1
		LIAR	0.49 ^a	0.51 ^b	0.50 ^c	–	0.50 ^h	–1
		FNN	0.48 ^a	0.57	0.50 ^c	0.53 ^{f,g}	–	0
		Avg. gain (%)	–4	0	–1	–1	0	–1
Coral	GloVe	Kaggle	–	0.70	0.57	0.63 ^{g,h}	0.61	1
		ISOT	0.57	–	0.51 ^{e,f}	0.61 ^{g,i,j}	0.54	–6
		AMT	0.53	0.49 ^{b,c}	–	0.55 ^{h,j,k}	0.50 ^l	–2
		LIAR	0.49 ^a	0.52 ^{b,d}	0.50 ^{e,g}	–	0.50 ^l	–1
		FNN	0.54 ^a	0.54 ^{c,d}	0.51 ^{f,g}	0.53 ^{j,k}	–	0
		Avg. gain (%)	–4	–1	–2	–1	0	–2

training, proposed independently by Tzeng *et al.* (2014) and Ganin and Lempitsky (2015). The initially proposed model consists not only of a feature extractor (neural network) and label predictor, but also of a domain classifier (critic), which is connected to the feature extractor. The network parameters are optimized to minimize the loss of the label classifier while maximizing the loss of the domain classifier. This forces the feature extractor to generate domain-invariant features. Saito *et al.* (2018) replaced the domain critic with dropout on the label predictor network. Hence, the label predictor acts as both the main classifier and domain classifier. Since the dropout is random, two instances of the label classifier with different nodes zeroed by the dropout may provide various predictions. The difference between these predictions can be viewed as a critic. During the adversarial training, the domain classifier tries to maximize this difference, while the feature generator tries to minimize it. Hence, the feature extractor is encouraged to generate more discriminative features for the target domain away from the region near the decision boundary.

- Virtual adversarial domain adaptation (VADA, Shu *et al.* 2018): a technique based on the cluster assumption, which states that the data that tend to form clusters and samples

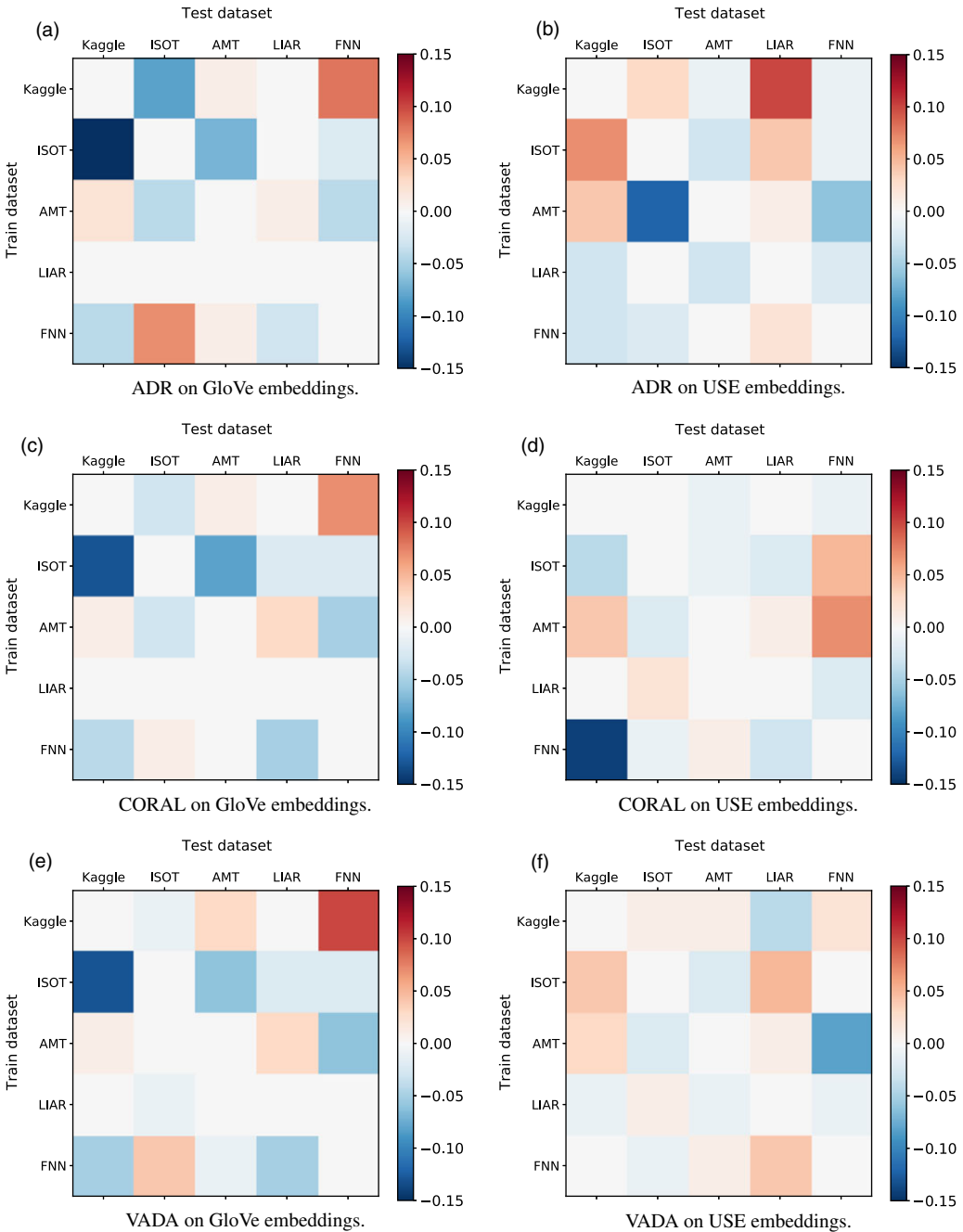


Figure 9. Improvement in accuracy achieved by different deep domain adaptation methods applied on two models: bidirectional-LSTM trained on GloVe embeddings and neural network using USE embeddings.

from the same cluster are likely to share the same label. It indicates that the decision boundaries should not cross the high-density region. The proposed model combines domain adversarial training with a penalty term that punishes violation of the cluster assumption.

Table 17. Ranking of methods for linguistic features

Method	Occurrences	General score	Normalized score
Feature intersection	20	35.0	1.75
GFK	20	25.0	1.25
XGBoost	20	22.0	1.10
SA	20	21.0	1.05
IRW	20	17.0	0.85

Table 18. Ranking of methods for GloVe features

Method	Occurrences	General score	Normalized score
BiLSTM	20	41.0	2.05
ADR	20	33.0	1.65
VADA	20	32.0	1.60
Coral	20	14.0	0.70

Table 19. Ranking of methods for USE features

Method	Occurrences	General score	Normalized score
Dense	20	23.0	1.15
SA	20	18.0	0.90
ADR	20	17.0	0.85
VADA	20	17.0	0.85
Coral	20	16.0	0.80
GFK	20	15.0	0.75
Linear SVC	20	14.0	0.70

9.1 Results of deep domain adaptation methods

The selected deep domain adaptation methods were applied to two models whose architectures are depicted in Figure 2. One of them utilized USE text embeddings, while the other was based on bi-directional LSTM using GloVe embeddings. In the context of domain adaptation, these models (except the part devoted to the final classification) are considered as features extractors. The performance of deep domain adaptation methods was evaluated for each source–target (train–test) pair of data sets. The entirety of one data set was used as the source, while the other was used as the target.

The adaptation methods were implemented in such a way that one batch of the source data and one batch of the target data were processed simultaneously. After each pair of batches, the model's loss was updated. Therefore, the batch sizes were adjusted in order to make the number of batches for both collections equal. The batch sizes were calculated according to the formula: $\lceil 0.9 \text{ dataset_size/num_of_batches} \rceil$ for source and $\lceil \text{dataset_size/num_of_batches} \rceil$

Table 20. Ranking of feature types

Features	Occurrences	General score	Normalized score
linguistic	100	68.0	0.68
GloVe	80	26.0	0.32
USE	140	26.0	0.18

for target. The factor 0.9 originates from using a 10-fold cross validation procedure. In our experiments, we set the parameter *num_of_batches* for 30. In the case of domain adaptation methods that were applied to the neural network based on USE embeddings, the number of epochs was set for 30. For bi-LSTM utilizing GloVe embeddings, we used 10 epochs due to the long training time.

The results of the deep domain adaptation methods are presented in Tables 15 and 16. Figure 9 illustrates the improvement achieved by applying different deep domain adaptation methods for each source–target pair of data sets. As a reference, we used the results that were obtained by the same classifiers, but without implementing any deep domain adaptation techniques.

On the whole, the domain adaptation methods did not provide any consistent improvement in classification accuracy for all pairs of data sets. In some cases, the results turned out to be lower than before domain adaptation. Such instances are marked in blue.

In the case of the model built on bidirectional LSTM architecture that takes GloVe embeddings as the input, the domain adaptation techniques provided some improvement for less than half of the source–target pairs. The overall gain for all the tested methods (ADR, CORAL, VADA) was close to zero.

The maps visualizing the gain in accuracy vary depending on the adaptation method and the model that was used; still, some common features can be observed.

First, all the investigated deep domain-adaptation methods improved classification accuracy for the two smallest data sets—AMT and FakeNewsNet—when Kaggle was used as a source. At the same time, the highest decrease in accuracy is observed for the ISOT-Kaggle pair, the two largest data sets. Secondly, when LIAR was used as a source, hardly any change was observed.

In the case of a model utilizing USE embeddings, the results were slightly better. They could be improved by selecting the number of epochs depending on the source–target pair and methods that are used.

Both VADA and adversarial dropout regularization contributed to improved accuracy when large data sets such as Kaggle and ISOT were used as a source.

To conclude, domain adaptation methods achieved higher improvement when applied to models utilizing USE embeddings.

10. Conclusions

10.1 Method ranking

Comparing all the results presented in our article is difficult: there are many combinations of training and test sets, and the situation is further complicated by the multiplicity of classification algorithms and feature spaces. To enable comparison, this section presents an aggregated view of our results. This is done by a ranking that summarizes the performance of methods presented in this article. We constructed these ranks as follows. For every train–test pair, we checked which three methods and feature space types worked best. We awarded three points for the best method or feature space, two points for the second-best score, and one point for the third-best score. The sum of these points is our general ranking score.

To account for a different number of occurrences, we also calculated how many times a given method and feature space was used, and for normalization we divided the general score by that number. This normalized score is arguably the best view of our ranking results.

We only evaluated three types of features: GloVe, USE, and linguistic features, as these were the most important feature space types for the different methods that were used.

Tables 17, 18, and 19 present rankings for methods applied to three feature types. Feature Intersection represents the classifier based on the intersection of the top features as in Section 7.2.

Table 20 compares all three feature types according to the ranking procedure. The results reveal linguistic features as the best performing feature type. Combined with the Feature Intersection classifier, it is likely the most versatile cross-domain approach for fake news detection.

10.2 General discussion

This paper investigated the possibility of designing a universal system for fake news detection purely from text that does not use fact-checking or knowledge bases. Promising preliminary results obtained for classifiers trained and tested on five publicly available fake news data sets (in-domain) turned out to be misleading. Simple cross-domain experiments revealed a significant decrease in accuracy: models trained on texts from one data set provided significantly lower results when evaluated on other data sets.

Therefore, we investigated a variety of methods to suppress this unwanted behavior. We compared domain adaptation techniques from three categories: feature selection and methods from the area of machine learning and deep learning. The feature selection approach is based on training models on a common set of features identified as important for several data sets. The goal of the cross-domain machine learning approaches tested in this paper is to address the phenomenon of data set shift. Deep learning methods are the newest ones; they originate from cross-domain scenarios and have been applied mostly in computer vision and graphics. To our knowledge, this paper is the first to compare all of these approaches on a fake news detection task.

We trained models on four types of features: bag-of-words vectors, 271 linguistic and psycholinguistic features, GLoVe embeddings, and text embeddings obtained from the Universal Sentence Encoder (USE).

In the case of the in-domain scenario, the best accuracy varied between 0.68 (LIAR) and 0.99 (Kaggle, ISOT). The accuracy depended on the data set size and text length. In order to illustrate the differences between the data sets, we compared the relationship between classification accuracy and the number of training samples. This analysis revealed that models learn differently not only on different data sets, but also when various text representations are used. This is most pronounced for AMT: only models trained on linguistic representations achieve reasonable results. Furthermore, the accuracy of models trained and evaluated on short single claims, such as those in LIAR, did not exceed 0.68. It seems that the veracity of such short texts can be accessed exclusively by fact-checking systems. This can be explained by the fact that the extraction of stylometric and psycholinguistic features from short statements can be prone to large errors.

The cross-domain scenario, in which models were trained on one data set and tested on another, revealed sharp drops in accuracy. An accuracy of over 0.9 becomes in many cases 0.6 or even 0.5. The best-performing solutions were LinearSVC on BOW followed by Bi-LSTM on GloVe and Dense on USE. Contrary to our initial expectations, linguistic features were the worst to generalize to other data sets.

Our feature selection results emphasized the role of adjectives and numbers in detecting fake news. Perhaps the most optimistic observation was that by using Mutual Information and the intersection of the top 100 linguistic features we can raise cross-domain performance of XGBoost models by 4%. Linguistic classifiers showed an overall greater improvement (4% averaged over all train–test pairs) than BOW (average gain equal to 1%).

Machine learning adaptation methods provided higher improvement for linguistic classifiers than for the ones based on USE embeddings. However, the obtained results varied significantly between the pairs of data sets. The highest overall improvement (averaged over all train–test pairs), equal to 3%, was achieved by subspace alignment applied to a classifier based on linguistic features. The success of adaptation methods on classifiers based on linguistic features may be ascribed to the initially good performance of this method followed by a considerable drop in performance in cross-domain settings. Good data set-specific results of these classifiers suggest the possibility to represent text features desired in the fake news detection task. Hence, domain adaptation methods were able to adjust features to fit different data sets.

Deep domain adaptation techniques turned out to be successful only when applied to some source–target pairs. Unfortunately, none of the tested deep domain adaptation techniques provided any improvement in the case of the model built upon a bidirectional LSTM architecture using GloVe embeddings as input; the average gain was close to zero.

The main goal of our paper was to explore the limits of cross-domain fake news detection. In summary, the conclusion from cross-domain scenarios is that the best versatility can be achieved when training models on the Kaggle data. This data set is the second largest in terms of the number of documents and the largest in terms of document size. Thanks to these factors, the accuracy for models trained on it is the highest, even when tested on other data sets. This holds for data that is very different in terms of topics, length and structure, including different definitions of “fakeness”.

The most promising techniques are based on linguistic features and either training models on the intersection of the top 100 features (4% average gain and the best performance according to the ranking approach) or the subspace alignment (4% average gain).

From a broader perspective, the lessons learned on the path towards versatile fake news detection are as follows. Due to extremely low recall, one should not consider automated fact-checking (this applies to systems similar to those submitted to the FEVERⁿ competition: based on Wikipedia and neural models for natural language inference) as a feasible option. Even as one of several modules, its impact will be marginal. In the role of a universally applicable system based on a single model, one can envision a model trained on the intersection of linguistic features.^o

One can extend it further by utilizing data set-specific results and designing a multi-model system. For instance, on the AMT data set linguistic features achieved an in-domain accuracy of 0.80. Articles similar to AMT (in terms of topics and length) could be handled by the AMT-trained model using linguistic features only, applying one of the two cross-domain adaptation methods mentioned previously. A similar procedure can lead to many dedicated models handling various types of texts, perhaps preserving the Kaggle-trained model as a default or back-off one. These are the hypothetical research directions that can be taken up in future studies but are not within the direct scope of our paper. Testing these ideas requires a new data set that includes samples of many possible forms of fake news.

It is worth noting that models without a fact-checking component may be vulnerable to attacks by adversarial generative models trained on how to modify fake or deceiving articles to make them appear like true ones.

Overall, overcoming the problem of differences between fake news data sets poses a major challenge that is much less studied than other areas in natural language processing, which makes our paper an important contribution.

Acknowledgements. This work was supported by the Poznan Supercomputing and Networking Center grant number 442.

ⁿ<https://fever.ai>.

^oThe superiority of linguistic features over fact-checking was also reported in Wawer, Wojdyga, and Sarzyńska-Wawer (2019). However, the data set was not fake news, but true and false statements in the general domain.

References

- Ahmed H., Traore I. and Saad S.** (2017). Detection of online fake news using N-gram analysis and machine learning techniques. In Traore I., Woungang I. and Awad A. (eds), *Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments*. Cham: Springer International Publishing, pp. 127–138.
- Allcott H. and Gentzkow M.** (2017). Social media and fake news in the 2016 election. *Journal of Economic Perspectives* 31(2), 211–36.
- Augenstein I., Lioma C., Wang D., Chaves Lima L., Hansen C., Hansen C. and Simonsen J.G.** (2019). MultiFC: A real-world multi-domain dataset for evidence-based fact checking of claims. 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)*, Hong Kong, China. Association for Computational Linguistics, pp. 4685–4697.
- Barrón-Cedeño A., Elsayed T., Nakov P., Da San Martino G., Hasanain M., Suwaileh R., Haouari F., Babulkov N., Hamdan B., Nikolov A., Shaar S. and Ali Z.S.** (2020). Overview of checkthat! 2020: Automatic identification and verification of claims in social media. In *Experimental IR Meets Multilinguality, Multimodality, and Interaction: 11th International Conference of the CLEF Association, CLEF 2020, Thessaloniki, Greece, September 22–25, 2020, Proceedings*. Berlin, Heidelberg: Springer-Verlag, pp. 215–236.
- Cer D., Yang Y., Kong S.-y., Hua N., Limtiaco N., St. John R., Constant N., Guajardo-Cespedes M., Yuan S., Tar C., Strope B. and Kurzweil R.** (2018). Universal sentence encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium. Association for Computational Linguistics, pp. 169–174.
- Chu C. and Wang R.** (2018). A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics, pp. 1304–1319.
- Cochran W.G.** (1950). The comparison of percentages in matched samples. *Biometrika* 37(3/4), 256–266.
- Coleman M. and Liau T.L.** (1975). A computer readability formula designed for machine scoring. *Journal of Applied Psychology* 60(2), 283.
- Conroy N.K., Rubin V.L. and Chen Y.** (2015). Automatic deception detection: Methods for finding fake news. *Proceedings of the Association for Information Science and Technology* 52(1), 1–4.
- Demšar J.** (2006). Statistical comparisons of classifiers over multiple data sets. *Journal of Machine Learning Research* 7(1), 1–30.
- Devlin J., Chang M.-W., Lee K. and Toutanova K.** (2019). 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)*, Minneapolis, Minnesota. Association for Computational Linguistics, pp. 4171–4186.
- Elsayed T., Nakov P., Barrón-Cedeño A., Hasanain M., Suwaileh R., Da San Martino G. and Atanasova P.** (2019). Overview of the clef-2019 checkthat! lab: Automatic identification and verification of claims. In Crestani F., Braschler M., Savoy J., Rauber A., Müller H., Losada D.E., Heinatz Bürki G., Cappellato L. and Ferro N. (eds), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*. Cham: Springer International Publishing, pp. 301–321.
- Fernando B., Habrard A., Sebban M. and Tuytelaars T.** (2013). Unsupervised visual domain adaptation using subspace alignment. In *Proceedings of the 2013 IEEE International Conference on Computer Vision, ICCV'13*. Washington, DC, USA: IEEE Computer Society, pp. 2960–2967.
- Flesch R.** (1948). A new readability yardstick. *Journal of Applied Psychology* 32(3), 221.
- Fu L., Nguyen T.H., Min B. and Grishman R.** (2017). Domain adaptation for relation extraction with domain adversarial neural network. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, Taipei, Taiwan. Asian Federation of Natural Language Processing, pp. 425–429.
- Ganin Y. and Lempitsky V.** (2015). Unsupervised domain adaptation by backpropagation. In *Proceedings of the 32nd International Conference on Machine Learning - Volume 37, ICML'15*, pp. 1180–1189. JMLR.org.
- Gong B., Shi Y., Sha F. and Grauman K.** (2012). Geodesic flow kernel for unsupervised domain adaptation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), CVPR'12*. Washington, DC, USA: IEEE Computer Society, pp. 2066–2073.
- Gunning R.** (1969). The fog index after twenty years. *Journal of Business Communication* 6(2), 3–13.
- Horne B. and Adali S.** (2017). This just in: Fake news packs a lot in title, uses simpler, repetitive content in text body, more similar to satire than real news. In *Proceedings of the International AAAI Conference on Web and Social Media*, pp. 759–766.
- Janicka M., Pszozna M. and Wawer A.** (2019). Cross-domain failures of fake news detection. *Computación y Sistemas* 23(3), 1089–1097.
- Jiang J. and Zhai C.** (2007). Instance weighting for domain adaptation in NLP. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic. Association for Computational Linguistics, pp. 264–271.
- Khan J.Y., Khondaker M.T.I., Afroz S., Uddin G. and Iqbal A.** (2021). A benchmark study of machine learning models for online fake news detection. *Machine Learning with Applications* 4, 100032.

- Kincaid J., Fishburne R., Rogers R. and Chissom B. (1975). Research branch report 8–75. Memphis: Naval Air Station.
- Kraskov A., Stögbauer H. and Grassberger P. (2004). Estimating mutual information. *Physical Review E* **69**, 066138.
- Leippold M. and Diggelmann T. (2020). Climate-fever: A dataset for verification of real-world climate claims. In *NeurIPS 2020 Workshop on Tackling Climate Change with Machine Learning*.
- Levi O., Hosseini P., Diab M. and Broniatowski D. (2019). Identifying nuances in fake news vs. satire: Using semantic and linguistic cues. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*, Hong Kong, China. Association for Computational Linguistics, pp. 31–35.
- Liu P., Qiu X. and Huang X. (2017). Adversarial multi-task learning for text classification. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada. Association for Computational Linguistics, pp. 1–10.
- Nakov P., Barrón-Cedeño A., Elsayed T., Suwaileh R., Márquez L., Zaghouni W., Atanasova P., Kyuchukov S. and Da San Martino G. (2018). Overview of the clef-2018 checkthat! lab on automatic identification and verification of political claims. In Bellot P., Trabelsi C., Mothe J., Murtagh F., Nie J.Y., Soulier L., SanJuan E., Cappellato L. and Ferro, N. (eds), *Experimental IR Meets Multilinguality, Multimodality, and Interaction*, Cham: Springer International Publishing, pp. 372–387.
- Nakov P., Corney D., Hasanain M., Alam F., Elsayed T., Barrón-Cedeño A., Papotti P., Shaar S. and Da San Martino G. (2021a). Automated fact-checking for assisting human fact-checkers. In Zhou, Z.-H. (ed), *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI-21*. International Joint Conferences on Artificial Intelligence Organization, pp. 4551–4558.
- Nakov P., Da San Martino G., Elsayed T., Barrón-Cedeño A., Mguev R., Shaar S., Alam F., Haouari F., Hasanain M., Babulkov N., Nikolov A., Shahi G.K., Struß J.M. and Mandl T. (2021b). The clef-2021 checkthat! lab on detecting check-worthy claims, previously fact-checked claims, and fake news. In *Advances in Information Retrieval: 43rd European Conference on IR Research, ECIR 2021, Virtual Event, March 28–April 1, 2021, Proceedings, Part II*. Berlin, Heidelberg: Springer-Verlag, pp. 639–649.
- Newman M.L., Pennebaker J.W., Berry D.S. and Richards J.M. (2003). Lying words: Predicting deception from linguistic styles. *Personality and Social Psychology Bulletin* **29**(5), 665–675.
- Pang B. and Lee L. (2004). A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, Barcelona, Spain, pp. 271–278.
- Pedregosa F., Varoquaux G., Gramfort A., Michel V., Thirion B., Grisel O., Blondel M., Prettenhofer P., Weiss R., Dubourg V., Vanderplas J., Passos A., Cournapeau D., Brucher M., Perrot M. and Duchesnay E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research* **12**, 2825–2830.
- Peng H., Long F. and Ding C. (2005). Feature selection based on mutual information: Criteria of max-dependency, max-relevance, and min-redundancy. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(8), 1226–1238.
- Pennebaker J., Boyd R., Jordan K. and Blackburn K. (2015). The development and psychometric properties of LIWC2015. Technical report, Austin, TX: University of Texas at Austin.
- Pennington J., Socher R. and Manning C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar. Association for Computational Linguistics, pp. 1532–1543.
- Pérez-Rosas V., Kleinberg B., Lefevre A. and Mihalcea R. (2018). Automatic detection of fake news. In *Proceedings of the 27th International Conference on Computational Linguistics*, Santa Fe, New Mexico, USA. Association for Computational Linguistics, pp. 3391–3401.
- Peters M.E., Neumann M., Iyyer M., Gardner M., Clark C., Lee K. and Zettlemoyer L. (2018). Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 2227–2237.
- Popat K., Mukherjee S., Yates A. and Weikum G. (2018). DeClarE: Debunking fake news and false claims using evidence-aware deep learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, Brussels, Belgium. Association for Computational Linguistics, pp. 22–32.
- Potthast M., Kiesel J., Reinartz K., Bevendorff J. and Stein B. (2018). A stylometric inquiry into hyperpartisan and fake news. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Melbourne, Australia. Association for Computational Linguistics, pp. 231–240.
- Przybyla P. (2020). Capturing the Style of Fake News. *Proceedings of the AAAI Conference on Artificial Intelligence* **34**(01), 490–497.
- Quionero-Candela J., Sugiyama M., Schwaighofer A. and Lawrence N.D. (2009). *Dataset Shift in Machine Learning*. Cambridge, Massachusetts: The MIT Press.
- Rashkin H., Choi E., Jang J.Y., Volkova S. and Choi Y. (2017). Truth of varying shades: Analyzing language in fake news and political fact-checking. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, Copenhagen, Denmark. Association for Computational Linguistics, pp. 2931–2937.
- Saito K., Ushiku Y., Harada T. and Saenko K. (2018). Adversarial dropout regularization. In *International Conference on Learning Representations*.

- Santos R., Pedro G., Leal S., Vale O., Pardo T., Bontcheva K. and Scarton C.** (2020). Measuring the impact of readability features in fake news detection. In *Proceedings of the 12th Language Resources and Evaluation Conference*, Marseille, France. European Language Resources Association, pp. 1404–1413.
- Schuster T., Shah D., Yeo Y.J.S., Roberto Filizzola Ortiz D., Santus E. and Barzilay R.** (2019). Towards debiasing fact verification models. 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)*, Hong Kong, China. Association for Computational Linguistics, pp. 3419–3425.
- Shu K., Mahudeswaran D., Wang S., Lee D. and Liu H.** (2020). Fakenewsnet: A data repository with news content, social context and dynamic information for studying fake news on social media. *Big Data* **8**, 171–188.
- Shu R., Bui H., Narui H. and Ermon S.** (2018). A DIRT-t approach to unsupervised domain adaptation. In *International Conference on Learning Representations*.
- Silva A., Luo L., Karunasekera S. and Leckie C.** (2021). Embracing domain differences in fake news: Cross-domain fake news detection using multi-modal data. *Proceedings of the AAAI Conference on Artificial Intelligence* **35**(1), 557–565.
- Smith E.A. and Senter R.** (1967). Automated readability index. *AMRL-TR. Aerospace Medical Research Laboratories*, pp. 1–14.
- Stammbach D. and Neumann G.** (2019). Team DOMLIN: Exploiting evidence enhancement for the FEVER shared task. In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, Hong Kong, China. Association for Computational Linguistics, pp. 105–109.
- Stone P.J., Dunphy D.C., Smith M.S. and Ogilvie D.M.** (1966). *The General Inquirer: A Computer Approach to Content Analysis*. Cambridge, MA: The MIT Press.
- Sun B. and Saenko K.** (2016). Deep coral: Correlation alignment for deep domain adaptation. In Hua G. and Jégou, H. (eds), *Computer Vision – ECCV 2016 Workshops*. Cham: Springer International Publishing, pp. 443–450.
- Tchechmedjiev A., Fafalios P., Boland K., Gasquet M., Zloch M., Zapilko B., Dietze S. and Todorov K.** (2019). Claimskg: A knowledge graph of fact-checked claims. In Ghidini C., Hartig O., Maleshkova M., Svátek V., Cruz I., Hogan A., Song J., Lefrançois M. and Gandon F. (eds), *The Semantic Web – ISWC 2019*. Cham: Springer International Publishing, pp. 309–324.
- Thorne J., Vlachos A., Christodoulopoulos C. and Mittal A.** (2018). FEVER: A large-scale dataset for fact extraction and VERification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, New Orleans, Louisiana. Association for Computational Linguistics, pp. 809–819.
- Thorne J., Vlachos A., Cocarascu O., Christodoulopoulos C. and Mittal A.** (eds) (2019). *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, Hong Kong, China. Association for Computational Linguistics.
- Tzeng E., Hoffman J., Zhang N., Saenko K. and Darrell T.** (2014). Deep domain confusion: Maximizing for domain invariance. CoRR, abs/1412.3474.
- Vosoughi S., Roy D. and Aral S.** (2018). The spread of true and false news online. *Science* **359**(6380), 1146–1151.
- Wadden D., Lin S., Lo K., Wang L.L., van Zuylén M., Cohan A. and Hajishirzi H.** (2020). Fact or fiction: Verifying scientific claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online. Association for Computational Linguistics, pp. 7534–7550.
- Wang W.Y.** (2017). “liar, liar pants on fire”: A new benchmark dataset for fake news detection. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, pp. 422–426.
- Wawer A., Wojdyga G. and Sarzyńska-Wawer J.** (2019). Fact checking or psycholinguistics: How to distinguish fake and true claims? In *Proceedings of the Second Workshop on Fact Extraction and VERification (FEVER)*, Hong Kong, China. Association for Computational Linguistics, pp. 7–12.
- Wilcoxon F.** (1945). Individual comparisons by ranking methods. *Biometrics Bulletin* **1**(6), 80–83.
- Wilson G. and Cook D.J.** (2018). A survey of unsupervised deep domain adaptation. CoRR, abs/1812.02849.
- Xu H., Caramanis C. and Mannor S.** (2009). Robustness and regularization of support vector machines. *Journal of Machine Learning Research* **10**, 1485–1510.
- Yang Y., Zheng L., Zhang J., Cui Q., Li Z. and Yu P.S.** (2018). Ti-cnn: Convolutional neural networks for fake news detection. arXiv preprint arXiv:1806.00749.
- Zellers R., Holtzman A., Rashkin H., Bisk Y., Farhadi A., Roesner F. and Choi Y.** (2019). Defending against neural fake news. In Wallach H., Larochelle H., Beygelzimer A., d’Alché-Buc F., Fox E. and Garnett, R. (eds), *Advances in Neural Information Processing Systems*, vol. 32. Curran Associates, Inc.
- Zhou X., Jain A., Phoha V.V. and Zafarani R.** (2020). Fake news early detection: A theory-driven model. *Digital Threats: Research and Practice* **1**(2), 1–25.