Deception Detection Within and Across Domains: Identifying and Understanding the Performance Gap

SUBHADARSHI PANDA, Hunter College, City University of New York, USA
SARAH ITA LEVITAN, Hunter College, City University of New York, USA

NLP approaches to automatic deception detection have gained popularity over the past few years, especially with the proliferation of fake reviews and fake news online. However, most previous studies of deception detection have focused on single domains. We currently lack information about how these single-domain models of deception may or may not generalize to new domains. In this work, we conduct empirical studies of cross-domain deception detection in five domains to understand how current models perform when evaluated on new deception domains. Our experimental results reveal a large gap between within and across domain classification performance. Motivated by these findings, we propose methods to understand the differences in performances across domains. We formulate five distance metrics that quantify the distance between pairs of deception domains. We experimentally demonstrate that the distance between a pair of domains negatively correlates with the cross-domain accuracies of the domains. We thoroughly analyze the differences in the domains and the impact of fine-tuning BERT based models by visualization of the sentence embeddings. Finally, we utilize the distance metrics to recommend the optimal source domain for any given target domain. This work highlights the need to develop robust learning algorithms for cross-domain deception detection that generalize and adapt to new domains and contributes toward that goal.

ACM Reference Format:
Subhadarshi Panda and Sarah Ita Levitan. 2023. Deception Detection Within and Across Domains: Identifying and Understanding the Performance Gap. 1, 1 (February 2023), 27 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Deception detection is an important goal of law enforcement, military and intelligence agencies, as well as commercial organizations. In recent years, automatic deception detection in text has gained popularity in the Natural Language Processing (NLP) community, and researchers have studied cues to deception in a diverse set of domains. These include detecting deception in news [34], online reviews [20], interview dialogues [16], trial testimonies [7], and in games [32]. These studies have been useful for identifying linguistic characteristics of deception, and for developing machine learning techniques to automatically detect deceptive language.

Although deception detection is a popular task in the NLP research community, and there is a strong interest in commercial applications of this work, there exists a large gap between deception models trained under laboratory conditions, and the performance level that is needed in real-world deception. Although researchers have in some cases obtained very strong performance at deception detection, these studies have focused on single domains, often using small datasets. We currently lack information about how small-scale, single-domain models of deception may or may

Authors' addresses: Subhadarshi Panda, Hunter College, City University of New York, USA, spanda@gradcenter.cuny.edu; Sarah Ita Levitan, Hunter College, City University of New York, USA, sarah.levitan@hunter.cuny.edu.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.
© 2023 Association for Computing Machinery.
XXXX-XXXX/2023/2-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

not generalize to real-world data and new domains. This work aims to fill this gap and addresses the following research questions: (1) How do current models of deception perform within domain and across domain? (2) When there are performance gaps between within and across domain deception detection, can we explain why they occur? (3) Can we leverage our understanding of these performance gaps to improve cross-domain deception detection?

The contributions of this work include: (1) an empirical study of language-based deception detection models and their performance both within and across five domains; (2) an analysis of factors that affect cross-domain deception detection performance, including the amount of source/target training data, and the effects of fine-tuning embeddings; (3) a comparison of five measures of domain similarity and how they relate to cross-domain deception detection performance; and (4) a proposed cross-domain classification model that leverages domain distance to outperform several baseline models. This work is critical for understanding and contextualizing the successes of deception detection models thus far and gaining insights about the unique challenges of deception detection. The insights gained from this work will motivate and inform the development of more robust models of deception.[1]

The paper is organized as follows: In section 2 we review related work in text-based deception detection. Section 3 describes the five corpora that we use in this work, and Section 4 details the results of our empirical study of within and across domain deception detection. In Section 5 we gain a deeper understanding of the classification results by visualizing and analyzing the embeddings representations that the models learn during training. In Section 6 we explore distance metrics to quantify the distances between domains, to understand the disparities between within and across domain classification. We evaluate the distance metrics in Section 7. Section 8 presents our proposed classification approach that leverages the notion of domain distance to improve cross-domain deception detection. Finally, we conclude in Section 9 and discuss ideas for future work.

2 RELATED WORK

There have been several important studies of linguistic cues to deception in a diverse set of domains. These include detecting deception in news [27, 34], online reviews [8, 20], interview dialogues [16], trial testimonies [7, 22], and in games [26, 32]. These studies have been critical for identifying specific linguistic characteristics of deception, and for building models to automatically detect deceptive language. There is growing evidence that a deception detection system trained on one domain performs poorly on other domains [10, 19, 24, 36]. However, this trend has not been systematically tested on a diverse set of deception detection datasets, and there has been little work done to understand the gap between in-domain and cross-domain performance. Recently, Glenski et al. [9] benchmarked model robustness for detecting deception, focusing on deceptive news online. They observed a drop in performance when evaluating on news data from different platforms (e.g. Reddit vs. Twitter). We build on this important work and empirically evaluate deception detection across multiple deception domains and tasks, going beyond the focus on deceptive news.

Some studies show that better performing deception systems are obtained using combinations of domains for training [2, 10, 19]. Capuozzo et al. [2], for instance, use 4 out 5 domains for training, while testing on the 5th domain. In this work, we compare models trained on combinations of domains with models that only use a single domain for training and another single domain for testing, similar to Pérez-Rosas and Mihalcea [24]. Our single source to single target setup enables us to compute how they are related to each other by computing the distance between them. The closest existing work to our work for quantifying distance between deception domains is by Fitzpatrick

[1]The data and the code will be released to the public upon publication.

Domain	Number of tokens				Number of samples	
	Mean	Std.	1%ile	99%ile	Truthful	Deceptive
Fake news [23]	324.50	692.35	78.58	1936.71	490	490
Open-domain deception [25]	10.59	5.19	5.00	31.00	3584	3584
Cross-cultural deception [24]	81.47	32.06	24.99	177.04	200	200
Deceptive opinion spam [20]	167.79	98.93	40.99	504.00	800	800
Liar liar pants on fire [34]	20.21	11.46	6.00	46.00	4507	8284

Table 1. Summary statistics for datasets from different domains along with distribution of truthful and deceptive classes.

and Bachenko [4], who estimate if a deception detection system can generalize to a new domain by using the frequency of the top 10 unigrams in the source and target domains. We draw inspiration from existing work and test five distance metrics to understand the underlying challenges to cross-domain deception detection.

Outside of the application to deception detection, domain adaptation has been successful in cross-domain sentiment analysis, where the model is tested on a domain it did not see during training. One example of such work is by Barnes et al. [1]. They propose to project the source and target word embeddings to a shared space based on a set of pivot words. The pivot words are the words that do not change their sentiment across domains, for example, the word *good* is a positive sentiment word across domains. Pivot words can be obtained from the pre-defined sentiment tokens [12] or can be computed using mutual information. For deception detection, identifying pivot words is challenging since associating words with deception is not straightforward. Moreover, since the method from Barnes et al. [1] is heavily reliant on source and target word embeddings, it is necessary to capture the concept of deception in the word embeddings for them to be meaningful, which is non-trivial. Domain adaptation has also been studied for hate speech detection in the work by Yin and Zubiaga [37], where the authors outline the various reasons why classifiers struggle to generalize across hate domains such as usage of non-standard grammar and vocabulary, and also discuss existing attempts at overcoming such challenges. On the other hand, deception detection is more nuanced, making it challenging even for humans.

Applying recent techniques such as BERT [3] for deception detection has been shown to be effective [5]. However, it is pointed out by Fornaciari et al. [5] that BERT alone does not capture the implicit knowledge of deception cues and therefore it is harder for BERT to generalize across domains. In this work we explore BERT models in depth, studying variations of BERT models and analyzing the learned embeddings representations to gain insights about the models as well as their limitations. More recent works such as by authors of Fornaciari et al. [6] have also demonstrated the challenges of cross-domain deception classification. Fornaciari et al. [6] show that deception datasets which are annotated by crowdsourcing show mismatch with real world deception datasets. Moreover, the crowdsourced datasets can be thought of as belonging to a different domain in comparison to the fake reviews published online, since they are significantly different from the real datasets. The models trained on crowdsourced datasets do not generalize to predicting detection on real datasets. We analyze datatasets obtained from a variety of sources, including crowdsourcing, to understand how they may or may not generalize to other forms of deception datasets.

3 DATA

We use five deception datasets from different domains in this work. These datasets were collected under different experimental settings and represent a diverse spread of deception domains. They

Domain	Train size	Test size
Fake news [23]	784	196
Open-domain deception [25]	5734	1434
Cross-cultural deception [24]	320	80
Deceptive opinion spam [20]	1280	320
Liar liar pants on fire [34]	10232	2559

Table 2. Training and test sizes of different domains. The test sets are balanced for all the domains other than Liar liar pants on fire, for which there are 902 truthful and 1657 deceptive samples.

were selected because they are all publicly available, and have been widely used for training and evaluating within-domain deception detection performance.

Fake news. This is a set of fake and legitimate news compiled via a combination of crowdsourcing and webscraping [23]. Legitimate news articles were obtained by scraping articles from mainstream US-based news websites, while fake news articles were generated by crowdworkers who were instructed to mimic reporting styles in the legitimate news articles. There are a total of 980 articles in this dataset.

Open-domain deception. This dataset consists of short, open-domain truths and lies obtained via crowdsourcing [25]. Crowdworkers were instructed to contribute seven lies and seven truths on any topics of their choosing, each consisting of a sengle sentence. They were asked to generate lies that were plausible. There are a total of 7168 sentences in this dataset.

Cross-cultural deception. This corpus consists of a set of deceptive and truthful essays about three topics: opinions on abortion, opinions on death penalty, and feelings about a best friend. The data was collected from participants in four different cultures: US, India, Mexico, and Romania [24]. To ensure consistency with the other datasets in this work, we focus on the portion of this dataset that was collected from US participants, which includes a total of 400 essays.

Deceptive opinion spam. This is a collection of truthful and deceptive hotel reviews of 20 Chicago hotels [20]. Truthful five star reviews were scraped from TripAdvisor, and corresponding deceptive reviews for the same hotels were generated by crowdworkers. The crowdworkers were instructed to generate fake yet realistic positive reviews for hotels which they had never visited. In total there are 1600 hotel reviews in this dataset.

Liar liar pants on fire. This corpus contains a set of short statements, mostly by politicians, in various contexts spanning across a decade [34]. The statements were scraped from PolitiFact, which assigns a human-provided veracity label for each statement. These labels include categories such as *pants-fire* (extremely false), *half-true, true* etc. We collapse the labels in the dataset to binary truthful vs. deceptive classes to align with the labels in the other datasets.[2] This dataset is by far the largest of the datsets used in this work; it has a total of 12,791 text samples.

Since each dataset has been curated using different data collection techniques and have different topics and styles, we consider each dataset to represent a different domain without loss of generality. The summary statistics of the datasets in each domain are shown in Table 1. As shown in the table, the datasets vary in text lengths. *Open domain deception* has the shortest texts, with an average of 10.59 tokens per text, followed by Liar, with an average of 20.21 tokens per text. *Fake news*

[2]We map 'true', 'mostly-true' to truthful class and 'false', 'half-true', 'barely-true', 'pants-fire' to the deceptive class.

includes much longer texts in the form of news articles, with an average of 324.5 tokens per text. The variety of domain styles in our collection of datasets, evidenced by the range in text lengths, is useful for evaluating the robustness of deception classifiers across a diverse set of corpora. Table 1 also shows the number of truthful and deceptive samples per dataset. 4 of the 5 datasets have perfectly balanced classes, while *Liar liar pants on fire* has approximately 35% truthful samples and 65% deceptive samples.

Since the above deception datasets do not come with pre-defined train-test splits, we perform a stratified splitting of the dataset of each domain into training and test splits with 80% of the data used for training and 20% used for testing. The size of each split is shown in Table 2. These train/test splits are used consistently across all experiments in this work to ensure a fair comparison of results across experiments.

4 WITHIN AND ACROSS DOMAIN DECEPTION CLASSIFICATION

We focus on the task of deception detection in a cross-domain setting. Training classification models to detect deception depends on the availability of labeled training data. Since labeled data is often not available or difficult to obtain in new domains, cross-domain deception detection would enable detection of deception in domains with no labeled data.

We formally define the cross-domain deception classification task as follows. Let (X_S^i, y_S^i) denote the i-th sentence-label pair in the source domain D_S. Let X_T^i denotes the i-th sentence in the target domain D_T. The task is to predict if sentences $\{X_T^i\}_{i=1}^{N_T}$ of D_T are deceptive or truthful using a classifier trained only using $\{X_S^i\}_{i=1}^{N_S}$, $\{y_S^i\}_{i=1}^{N_S}$ and $\{X_T^i\}_{i=1}^{N_T}$, where N_S and N_T are the number of sentences in D_S and D_T respectively.

We conduct four sets of classification experiments and present the results in the subsections below. First, in Section 4.1, we establish baseline performance at within and cross-domain deception classification using two well-established NLP models: logistic regression [14, 34] using unigram features, and BERT [3].[3] Next, we train variations of the BERT model and explore the effects of freezing vs. unfreezing weights in Section 4.2. We then consider the impact of the amount of source and target training data available, comparing zero-shot, few-shot, and full-shot experimental setups in Section 4.3. Finally, we train models using combinations of source domains and compare them with singe source domain models in Section 4.4. These thorough experiments contribute substantially to our understanding of cross-domain deception detection.

4.1 Baselines

Source domain ↓ Target domain →	**FN**		**ODD**		**CCD**		**DOS**		**LLPF**	
Features used →	Word	POS	Word	POS	Word	POS	Word	POS	Word	POS
FN	<u>0.633</u>	<u>0.704</u>	0.490	0.528	0.525	0.500	0.516	0.556	0.547	0.501
ODD	0.485	0.510	<u>0.605</u>	<u>0.554</u>	0.375	0.475	0.550	0.509	0.551	0.508
CCD	0.561	0.510	0.508	0.515	<u>0.562</u>	<u>0.438</u>	0.487	0.491	0.615	0.472
DOS	0.515	0.582	0.503	0.499	0.450	0.575	<u>0.887</u>	<u>0.669</u>	0.360	0.470
LLPF	0.510	0.500	0.498	0.504	0.487	0.512	0.516	0.506	<u>0.653</u>	<u>0.650</u>

Table 3. Cross-domain accuracies for deception detection using the logistic regression model (two variants: word features and POS features). In-domain accuracies are also shown and underlined. (**FN**: Fake news, **ODD**: Open domain deception, **CCD**: Cross-cultural deception, **DOS**: Deceptive opinion spam, **LLPF**: Liar liar pants on fire)

We begin by establishing a baseline model for within and cross-domain deception detection. This is essential for understanding how a simple model performs at this task, and for comparing

[3]https://huggingface.co/blog/bert-101

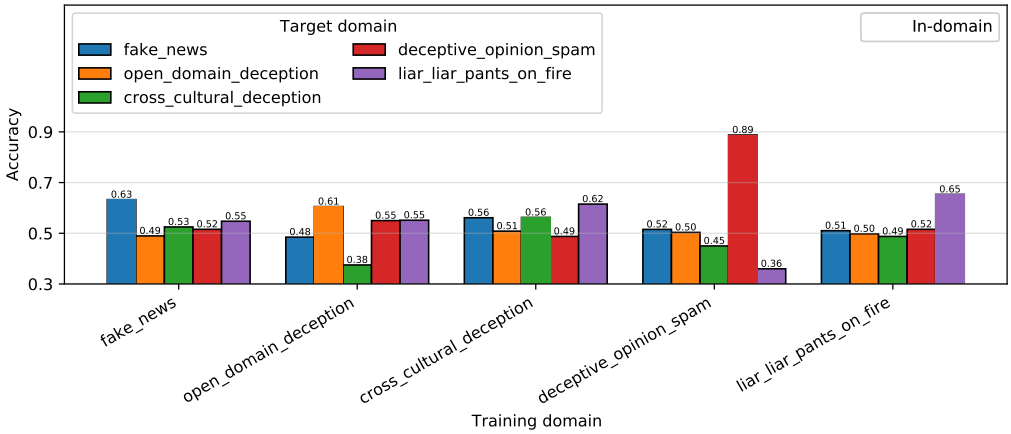

Fig. 1. In-domain and cross-domain accuracies using logistic regression model with word features.

with more complex models. Although customized state-of-the-art models have been developed for each of the datasets that we use in this work, the models are not typically released to the public, nor are specific train/tests splits published, so the results may not be directly reproducible. We first establish general baseline models that are consistently applied across all datasets, ensuring reproducibility and comparability. We train a logistic regression model using unigram TF-IDF features, a standard baseline for many NLP applications [13, 14, 17, 34]. The input to the logistic regression model is a vector for each sample of data. For a given training dataset, term frequencies and inverse document frequencies are computed to represent each sentence in the dataset as a TF-IDF vector. This TF-IDF vector is the input to the logistic regression classifier. The model is trained using the sentence-label pairs in the source domain D_S and the trained models are used to predict if the sentences in the target domain D_T are deceptive or truthful. To compare cross-domain performance and within-domain performance, the model is also trained and evaluated using each domain as both the source and target (using train and test partitions of the data).

The unigram TF-IDF features are commonly used in NLP tasks, including text-based deception detection. Logistic regression is a linear model that has a linear layer followed by a sigmoid layer. We tokenize the text and then lemmatize the tokens, both using spaCy [11], an open-source library for NLP in Python. For all the experiments, we used a 10% random split of the source domain training data as the development data. We tuned the C-parameter of the logistic regression model across the values {1, 2, 3, 4, 5} using the development accuracy.

Following previous work which shows the usefulness of part-of-speech (POS) features for deception detection [25], we also considered the POS tag of each token instead of the token itself, and obtained TF-IDF unigram features where the unigram tokens are POS tags themselves. We show the cross-domain accuracies for all source-target domain pairs in Table 3 for both word features and POS features. The results for within-domain deception detection are underlined in the table.

We observe in Table 3 that for any given target domain, the in-domain accuracies are generally higher than the cross-domain accuracies. This finding is consistent with observations made by Glenski et al. [9]. The above observation holds for both variants: using words as well as POS tags as features. In some cases, the gap between within and across domain performance is egregious. For example, a logistic regression model trained with word features on Deceptive Opinion Spam (DOS)

Source domain → Target domain	Log. reg.	BERT (Freeze)
fake news → fake news	**0.633**	0.541
fake news → open domain deception	0.49	**0.512**
fake news → cross cultural deception	0.525	**0.575**
fake news → deceptive opinion spam	**0.516**	0.5
fake news → liar liar pants on fire	**0.547**	0.353
open domain deception → open domain deception	**0.605**	0.573
open domain deception → fake news	0.485	**0.52**
open domain deception → cross cultural deception	0.375	**0.488**
open domain deception → deceptive opinion spam	**0.55**	0.5
open domain deception → liar liar pants on fire	0.551	**0.559**
cross cultural deception → cross cultural deception	**0.562**	0.5
cross cultural deception → fake news	**0.561**	0.5
cross cultural deception → open domain deception	**0.508**	0.507
cross cultural deception → deceptive opinion spam	0.488	**0.5**
cross cultural deception → liar liar pants on fire	**0.615**	0.36
deceptive opinion spam → deceptive opinion spam	**0.887**	0.806
deceptive opinion spam → fake news	0.515	**0.52**
deceptive opinion spam → open domain deception	**0.503**	0.502
deceptive opinion spam → cross cultural deception	0.45	**0.475**
deceptive opinion spam → liar liar pants on fire	0.36	**0.48**
liar liar pants on fire → liar liar pants on fire	**0.653**	0.646
liar liar pants on fire → fake news	0.51	**0.515**
liar liar pants on fire → open domain deception	0.498	**0.5**
liar liar pants on fire → cross cultural deception	0.488	**0.5**
liar liar pants on fire → deceptive opinion spam	**0.516**	0.5
Average	**0.536**	0.517

Table 4. Cross-domain accuracies using logistic regression unigram features vs BERT with frozen weights. In-domain accuracies are also shown and underlined.

has a within domain accuracy of 0.89, while the cross-domain performance of a model trained on DOS ranges from 0.360-0.515 for the four other target domains. Further, the cross-domain performance of models trained on other domains and tested on DOS ranges from 0.487-0.550. Although the DOS model has very strong within domain performance and is a useful model of deceptive hotel reviews, it is clearly not a robust model of deception and cannot generalize to other deception domains. These experiments highlight the need to benchmark models of deception across domains in order to provide proper context to interpret model performance. Further, our results call attention to the danger of applying deception detection models to new domains without properly evaluating their appropriateness to the target deception task. The scores are very close to each other when using either unigram word features or POS features. We focus on unigram word features for further experiments.

While a logistic regression model with n-gram features is a standard baseline for NLP tasks, it is not generally a state-of-the-art model. In our next experiments, we applied a state-of-the-art NLP model to establish a stronger baseline for within and cross-domain deception detection. Bidirectional Encoder Representations from Transformer (BERT) [3] has been shown to achieve promising results in a wide range of text classification tasks such as GLUE [33], and therefore we

explore the performance of BERT for deception detection. BERT uses self attention layers which encode a given sequence as set of vectors where each token's vector is influenced by other tokens. The self attention mechanism enables the model to learn contexualized representations for a given input sequence. BERT is pre-trained using self supervised learning from vast amounts of text data. Then the model can be used for downstream tasks which have annotated training data. Similar to the logistic regression based experiments, we used a 10% random split of the source domain training data as the development data. For deception classification, we trained a BERT-based sequence classification model by freezing the BERT parameters and tuning the multilayer perceptron (MLP) added on top.[4] For training the BERT-based model we used the Adam optimizer [15] with a learning rate of 0.003. The training was stopped when the development accuracy did not improve for 5 consecutive epochs. In all experimental setups, we tested on the target domain's test data.

We compare the performance of the BERT baseline with our previous logistic regression unigram baseline. Since we used development data to decide early stopping when training BERT, we also used the same amount of development data to tune the C-parameter in the logistic regression model for a fair comparison between BERT vs. logistic regression.

The results of the comparison of cross-domain accuracies of BERT vs logistic regression using unigram features are shown in Table 4. As shown in the table, the BERT based accuracies are better in 60% (12 out of 20) cross-domain experiments. However, we observe that surprisingly the in-domain results are always better using logistic regression. Additionally, the average performances across all domain pairs are very close to each other: 53.56% for logistic regression and 51.73% for BERT frozen model. Although BERT has been shown to outperform baselines like logistic regression by wide margins in other NLP tasks, we found that the margin of improvement was minimal for cross-domain deception detection.

We compare our in-domain baseline results with previously reported results for each dataset. In the cases where a comparison is not straightforward, we note the differences in the experimental setups.

(1) For the *Fake news* domain, Pérez-Rosas et al. [23] build a fake news classifier using data from the following topics: sports, business, entertainment, politics, technology, and education. They build another fake news classifier for the celebrity news. They test their classifiers on different news topics separately instead of combining all the news topic together. On the other hand, in this work we combine all the news topics together (that is, we consider all the news topics to fall under the *Fake news* domain) for training and testing classifiers. This makes their results not directly comparable with ours. Their best accuracies are in the range of 74% - 76%, whereas our best baseline accuracy is 78.6% (see Table 5).

(2) For the *Open domain deception* domain, the deception classification accuracy is reported to be 69.5% using a SVM model with part-of-speech features by Pérez-Rosas and Mihalcea [25]. In our experiments, we obtain a baseline accuracy score of 64.2% using a BERT classifier (see Table 5). While our score is lower, we note that the randomness in splitting of the dataset into train, development and test splits makes the scores not strictly comparable.

(3) For the *Cross-cultural deception* domain, Pérez-Rosas and Mihalcea [24] report in-domain deception classification scores by training and testing classifiers on individual topics (abortion, best friend, death penalty) or individual locales (English-US, English-India, Spanish-Mexico). However, in this work we consider only the English portion of the dataset and combine all the topics and locales in English together to train and test deception classifiers. Therefore, our results are not directly comparable to those by Pérez-Rosas and Mihalcea [24]. While

[4] `bert-base-uncased` model in Transformers library [35].

previously reported accuracies range from 63% - 73%, our best baseline score is 61.3% (see Table 5).

(4) For the *Deceptive opinion spam* domain, the best previously reported accuracy is 92.8% by Ren and Zhang [29], whereas our best baseline accuracy is 90.9% (see Table 5). We note that these scores are not strictly comparable because of the randomness in splitting of the dataset into train, development and test splits.

(5) For the *Liar liar pants on fire* domain, the scores reported by Wang [34] are based on classification into six categories: pants-fire, false, barely-true, half-true, mostly-true, and true. However in our setup we classify into two categories: truthful and deceptive. Therefore the results are not directly comparable. The six-way classification accuracy as reported by Wang [34] is 27%, while our best baseline accuracy for two-way classification is 67.4% (see Table 5).

Having established two baseline models for deception classification, we next explore the problem more deeply. We explore variations of the BERT model, and also examine the effects of different amounts of training data on classification performance, and as detailed in the subsections below.

4.2 Variations of BERT models

In our next experiment, we compare two variations of BERT models for deception classification: BERT Devlin et al. [3] and DistilBERT Sanh et al. [30]. BERT (base model) is a large language model with 110 M parameters trained using masked language modeling on the BookCorpus and Wikipedia data. While it is very useful for many NLP tasks, BERT requires extensive computational and memory resources due it its large size. Because of these concerns, DistilBERT was developed as a smaller, more lightweight alternative to BERT. It is a distilled BERT model with 66 M parameters. In spite of having a small size, DistilBERT has been shown to be effective on downstream classification tasks by retaining 97% of BERT performance on General Language Understanding Evaluation (GLUE) benchmark tasks [33].

To use the above BERT-based models for classification, we add a classification head on top for deception classification. In addition to comparing BERT vs. DistilBERT model performance for deception classification, we explore the effects of freezing vs. unfreezing model parameters on classification performance. To do this, we train two versions of each model: 1. by freezing the pre-trained weights, and 2. by unfreezing the pre-trained weights. When the pre-trained weights are frozen, only the classification layers' weights are trained. On the other hand, when the pre-trained weights are made trainable, all the weights are trained. The unfrozen version of the model generally requires more time to train an epoch in comparison to when the weights are frozen.

The classification results of using different BERT models under different conditions (freezing/unfreezing the pre-trained weights) are shown in Table 5. We observe that BERT outperforms DistilBERT in a majority of cases. This is intuitive because it is a larger model with more parameters, but it also has the disadvantage of longer training time than DistilBERT. When comparing frozen vs. unfrozen models, we observe that unfreezing the pre-trained weights leads to better cross-domain generalization. This finding is consistent with that of Fornaciari et al. [5] who demonstrate that BERT alone does not capture the implicit knowledge of deception cues. It is necessary to fine-tune BERT including the pre-trained weights on deception detection data to learn the deception cues. Upon freezing the BERT layers, we find the the cross-domain classification is close to random guess in most source domain $\rightarrow$ target domain classifications.

4.3 Impact of amount of source and target domain training data

So far all of our experiments were conducted in a fully cross-domain setup, where we train our models on a source domain and evaluate the performance of the model in a new target domain.

Source domain → Target domain	BERT (Freeze)	BERT (Unfreeze)	Distilbert (Freeze)	Distilbert (Unfreeze)
fake news → fake news	0.541	**0.786**	0.633	0.77
fake news → open domain deception	0.512	**0.518**	0.512	0.501
fake news → cross cultural deception	**0.575**	0.5	0.488	0.488
fake news → deceptive opinion spam	0.5	**0.572**	0.431	0.503
fake news → liar liar pants on fire	0.353	**0.62**	0.419	0.562
open domain deception → open domain deception	0.573	**0.642**	0.579	0.631
open domain deception → fake news	0.52	0.474	**0.566**	0.561
open domain deception → cross cultural deception	**0.488**	0.4	0.488	0.375
open domain deception → deceptive opinion spam	0.5	0.478	**0.512**	0.431
open domain deception → liar liar pants on fire	0.559	0.581	**0.585**	0.557
cross cultural deception → cross cultural deception	0.5	**0.613**	0.575	0.6
cross cultural deception → fake news	0.5	**0.566**	0.551	0.531
cross cultural deception → open domain deception	0.507	0.504	**0.51**	0.499
cross cultural deception → deceptive opinion spam	**0.5**	0.456	0.406	0.359
cross cultural deception → liar liar pants on fire	0.36	0.501	0.562	**0.612**
deceptive opinion spam → deceptive opinion spam	0.806	**0.909**	0.753	0.891
deceptive opinion spam → fake news	0.52	0.52	0.515	**0.566**
deceptive opinion spam → open domain deception	**0.502**	0.5	0.494	0.5
deceptive opinion spam → cross cultural deception	0.475	**0.55**	0.462	0.462
deceptive opinion spam → liar liar pants on fire	0.48	0.453	0.382	**0.518**
liar liar pants on fire → liar liar pants on fire	0.646	**0.674**	0.662	0.663
liar liar pants on fire → fake news	**0.515**	0.5	0.51	0.51
liar liar pants on fire → open domain deception	0.5	**0.504**	0.499	0.497
liar liar pants on fire → cross cultural deception	**0.5**	0.5	0.5	0.5
liar liar pants on fire → deceptive opinion spam	0.5	**0.506**	0.488	0.5

Table 5. Cross-domain accuracies using different BERT based classification models. In-domain accuracies are also shown and underlined.

We compared the cross-domain to within-domain results and identified a large performance gap: model performance sharply decreases when tested in a new domain. In our next experiments, we aimed to gain a deeper understanding of model performances in cross-domain deception detection. Specifically, we were interested in exploring the impact of the amount of source and target training data available. While models performed poorly when training and testing in completely different domains, how would the results change if we use some target data to train our models?

To enable this analysis, we perform experiments in the following setups, which are defined by the amount of target domain training data used. In all the setups, the testing is done on the target domain test set.

Zero shot. In the zero shot setup, the training data consists of only source domain training data; zero instances of the target data are used in training. (All of the results reported in Sections 4.1 and 4.2 used the zero shot setup.)

Few shot. In the few shot setup, the training data consists of all source domain training data and x% of the target domain training data, sampled uniformly at random. We set x to 33% and 67%. We note that only target training data is combined with source training data – the target test data remains unseen at training time. Training in the few shot setup enables the model to see a few samples from the target domain during training. Therefore, it is expected that the few shot setup will achieve better generalization across domains.

Source domain → Target domain	Setup	Log. reg.	BERT (Freeze)	BERT (Unfreeze)	Distilbert (Freeze)	Distilbert (Unfreeze)
fake news → open domain deception	zero	0.49	0.512	**0.518**	0.512	0.501
	few 33%	0.584	0.537	**0.624**	0.515	0.622
	few 67%	0.589	0.526	0.635	0.546	**0.651**
	full	0.603	0.548	0.63	0.607	**0.635**
fake news → cross cultural deception	zero	0.525	**0.575**	0.5	0.488	0.488
	few 33%	0.612	0.5	**0.638**	0.575	0.612
	few 67%	0.575	0.588	0.612	0.538	**0.625**
	full	0.55	0.5	**0.662**	0.525	0.538
fake news → deceptive opinion spam	zero	0.516	0.5	**0.572**	0.431	0.503
	few 33%	0.822	0.669	**0.825**	0.622	0.781
	few 67%	**0.834**	0.734	0.772	0.706	0.778
	full	**0.884**	0.738	0.869	0.797	0.872
fake news → liar liar pants on fire	zero	0.547	0.353	**0.62**	0.419	0.562
	few 33%	**0.654**	0.551	0.633	0.638	0.641
	few 67%	0.644	0.565	0.598	**0.659**	0.623
	full	0.644	0.646	0.621	**0.663**	0.61
open domain deception → fake news	zero	0.485	0.52	0.474	**0.566**	0.561
	few 33%	0.52	0.541	**0.679**	0.602	0.633
	few 67%	0.551	0.628	**0.755**	0.612	0.694
	full	0.566	0.5	**0.745**	0.582	0.714
open domain deception → cross cultural deception	zero	0.375	**0.488**	0.4	0.488	0.375
	few 33%	0.462	0.5	**0.538**	0.512	0.45
	few 67%	0.462	0.488	**0.575**	0.5	0.5
	full	0.438	0.5	**0.625**	0.575	0.525
open domain deception → deceptive opinion spam	zero	**0.55**	0.5	0.478	0.512	0.431
	few 33%	0.75	0.728	0.65	0.675	**0.766**
	few 67%	0.794	0.712	0.781	0.631	**0.844**
	full	0.819	0.744	**0.894**	0.75	0.828
open domain deception → liar liar pants on fire	zero	0.551	0.559	**0.581**	0.585	0.557
	few 33%	0.64	0.409	0.634	**0.642**	0.621
	few 67%	0.62	0.364	0.622	**0.658**	0.658
	full	0.661	0.647	0.631	**0.662**	0.653
cross cultural deception → fake news	zero	0.561	0.5	**0.566**	0.551	0.531
	few 33%	0.663	0.52	0.679	0.587	**0.709**
	few 67%	0.658	0.561	0.51	0.597	**0.781**
	full	0.663	0.607	**0.796**	0.597	0.709
cross cultural deception → open domain deception	zero	0.508	0.507	0.504	**0.51**	0.499
	few 33%	0.589	0.504	0.573	0.538	**0.616**
	few 67%	0.572	0.5	**0.646**	0.515	0.637
	full	0.596	0.547	**0.655**	0.586	0.602
cross cultural deception → deceptive opinion spam	zero	0.488	**0.5**	0.456	0.406	0.359
	few 33%	0.834	0.759	**0.838**	0.688	0.775
	few 67%	0.859	0.647	**0.862**	0.681	0.841
	full	**0.891**	0.697	0.853	0.794	0.8
cross cultural deception → liar liar pants on fire	zero	**0.615**	0.36	0.501	0.562	0.612
	few 33%	0.651	0.572	0.633	0.626	**0.652**
	few 67%	0.654	0.424	0.648	**0.656**	0.654
	full	0.655	0.647	**0.668**	0.66	0.664

Table 6. Cross-domain accuracies using different amounts of target domain training data (zero, few and full shot setups) and using different classification models.

Full shot. The full shot setup is a special case of the few shot setup where $x = 100\%$. In other words, all the source domain training data as well as target domain training data are used for training the model.

The cross-domain accuracies in different setups are shown in Tables 6 and 7. First, we observe that the cross-domain classification improves when using some or all the samples in the target domain. The classification accuracies generally increase as we use 33%, 67% and 100% of the training samples in the target domain. For instance, for open domain deception → cross cultural deception classification, the accuracy using BERT fine-tuning in zero shot condition is 40%. It increases to 53.8% and 57.5% in few shot conditions when 33% and 67% target domain samples are also added to the training data. Upon adding all the target domain samples to the training data, the accuracy goes up to 62.5%. We also observe that this trend is similar across classifiers: we see this in Tables 6 and 7 for logistic regression model and all different variants of BERT based classifiers. These experiments highlight the value of even a small amount of target training data in improving cross-domain deception detection. Some model performances improve dramatically from zero shot to few shot.

Source domain → Target domain	Setup	Log. reg.	BERT (Freeze)	BERT (Unfreeze)	Distilbert (Freeze)	Distilbert (Unfreeze)
deceptive opinion spam → fake news	zero	0.515	0.52	0.52	0.515	**0.566**
	few 33%	0.638	0.526	**0.735**	0.536	0.673
	few 67%	0.628	0.668	**0.75**	0.658	0.699
	full	0.571	0.546	**0.755**	0.628	0.73
deceptive opinion spam → open domain deception	zero	**0.503**	0.502	0.5	0.494	0.5
	few 33%	0.554	0.524	**0.6**	0.535	0.597
	few 67%	0.579	0.531	**0.632**	0.575	0.624
	full	0.59	0.52	**0.625**	0.596	0.616
deceptive opinion spam → cross cultural deception	zero	0.45	0.475	**0.55**	0.462	0.462
	few 33%	0.55	0.5	**0.562**	0.488	0.538
	few 67%	0.525	0.55	0.588	0.488	**0.612**
	full	0.412	0.538	0.612	**0.662**	0.5
deceptive opinion spam → liar liar pants on fire	zero	0.36	0.48	0.453	0.382	**0.518**
	few 33%	0.635	0.408	0.595	0.603	**0.606**
	few 67%	0.639	0.505	0.606	**0.659**	0.636
	full	0.637	0.647	0.634	**0.661**	0.633
liar liar pants on fire → fake news	zero	0.51	**0.515**	0.5	0.51	0.51
	few 33%	0.551	0.531	0.689	0.592	**0.719**
	few 67%	0.587	0.612	**0.77**	0.658	0.714
	full	0.582	0.5	**0.699**	0.597	0.648
liar liar pants on fire → open domain deception	zero	0.498	0.5	**0.504**	0.499	0.497
	few 33%	0.563	0.539	0.55	0.509	**0.584**
	few 67%	0.576	0.501	**0.633**	0.514	0.623
	full	0.586	0.529	0.607	0.562	**0.614**
liar liar pants on fire → cross cultural deception	zero	0.488	**0.5**	0.5	0.5	0.5
	few 33%	0.512	0.475	**0.562**	0.475	0.512
	few 67%	0.5	0.512	**0.55**	0.512	0.538
	full	0.475	0.5	**0.612**	0.538	0.588
liar liar pants on fire → deceptive opinion spam	zero	**0.516**	0.5	0.506	0.488	0.5
	few 33%	0.762	0.562	**0.816**	0.697	0.803
	few 67%	0.809	0.578	**0.834**	0.556	0.828
	full	0.828	0.744	0.866	0.75	**0.859**

Table 7. Cross-domain accuracies using different amounts of target domain training data (zero, few and full shot setups) and using different classification models.

Test domain	BERT freeze		BERT unfreeze		DistilBERT freeze		DistilBERT unfreeze	
	single-source	multi-source	single-source	multi-source	single-source	multi-source	single-source	multi-source
fake news	0.541	0.531	0.786	**0.679**	0.633	0.495	0.77	0.653
open domain deception	0.573	0.504	0.642	**0.630**	0.579	0.504	0.631	0.590
cross cultural deception	0.5	0.500	0.613	0.487	0.575	0.500	0.6	**0.525**
deceptive opinion spam	0.806	0.500	0.909	**0.722**	0.753	0.500	0.891	0.666
liar liar pants on fire	0.646	0.647	0.674	0.643	0.662	**0.662**	0.663	0.649

Table 8. Accuracies of different BERT based classifiers trained in single-source setting vs multi-source setting. For each test domain, the best single-source accuracy is underlined and the best multi-source accuracy is shown in bold.

For example, open domain deception → deceptive opinion spam performance improves from 0.478 to 0.781 between zero and few shot, and as high as 0.893 in the full shot setup. Interestingly, in some cases, model performance is optimized in a few shot setup rather than full shot. For example, for liar liar pants on fire → fake news, the best performance of 0.719 is obtained using DistilBERT unfrozen in the few 33% setup; this performance is better than the full shot setup for the same model which achieves an accuracy 0.648.

In this analysis, we used two domains only: the source domain and the target domain. Next, we explored the effects of training with multiple source domains and evaluating with single target domains, as described below.

4.4 Multi-source training

In this section we present results of another formulation of the classification problem: multi-source training. Instead of training a model using a single source domain, we explore models trained with training data from multiple source domains. We combine the training datasets from all the 5 domains to obtain a concatenated multi-source training data. We also combine the development datasets from all the 5 domains and obtain a concatenated development data. In the multi-source setup, we focus on training and evaluating BERT-based models, since they were found to outperform logistic regression in our prior experiments. We train a single BERT-based classifier using the concatenated training data and determine early stopping based on the model's accuracy on the concatenated development data. For evaluation, we predict the labels of each domain's test data separately and report the accuracy scores in Table 8.

Table 8 shows the results of four BERT-based models trained on single-source and multi-source training data: BERT frozen, BERT unfrozen, DistilBERT frozen, and DistilBERT unfrozen. For the single-source training data setting, the training and testing are done on the same domain. Each row in Table 8 displays the results of the model for a particular target domain. Somewhat surprisingly, we find that the performance on the test sets for multi-source training are worse than the single source - single target training. For example, the multi-source test accuracy for deceptive opinion spam is 72.2% when BERT is fine-tuned without freezing the BERT layers, whereas the single-source test accuracy is 90.9%. Interestingly, the accuracy is as high as 89.4% when only open domain deception data and deceptive opinion spam data is used for training in the full shot setup (see Table 7). This indicates that the accuracy of detecting deceptive opinion spam drops when using all the domains for training. We hypothesize that the nature of deception varies significantly across domains, and the features of deceptive language in one domain may not be associated with deception in all other domains. Therefore when samples from all the domains are combined together, the possibly conflicting or inconsistent deception signals from different domains may make it harder for the model to learn useful deception cues for a specific target domain. On the other hand, in the full shot experiments shown in Tables 6 and 7, we only combine the target training data with a single source domain. If the target and source data are a good match, the model is able to learn reliable cues to deception, which results in the best deception classification performance. In order to understand the BERT-based models and how they may or may not generalize to new domains, we examine the embeddings representations that are learned by the models in the next section.

5 UNDERSTANDING THE CLASSIFICATION RESULTS VIA EMBEDDINGS VISUALIZATIONS

Section 4 presented several classification experiments that demonstrated the performance gap between within and cross-domain deception detection, and compared several classification models and the tradeoffs between them. The results in the previous section answered many research questions, but also raised several new questions.

We observed substantial variation in cross-domain performance across different source/target domain pairs: some pairs appear to be a good fit for cross-domain deception detection, while others perform quite poorly. This motivates the question, why are some source-target pairs more effective for cross-domain classification, and why are other pairs more difficult to classify?

We also observed that fine-tuning the BERT-based models, i.e. training them with unfrozen weights that are updated during training, significantly improves performance. Why is fine-tuning an important part of the training process to improve cross-domain deception detection?

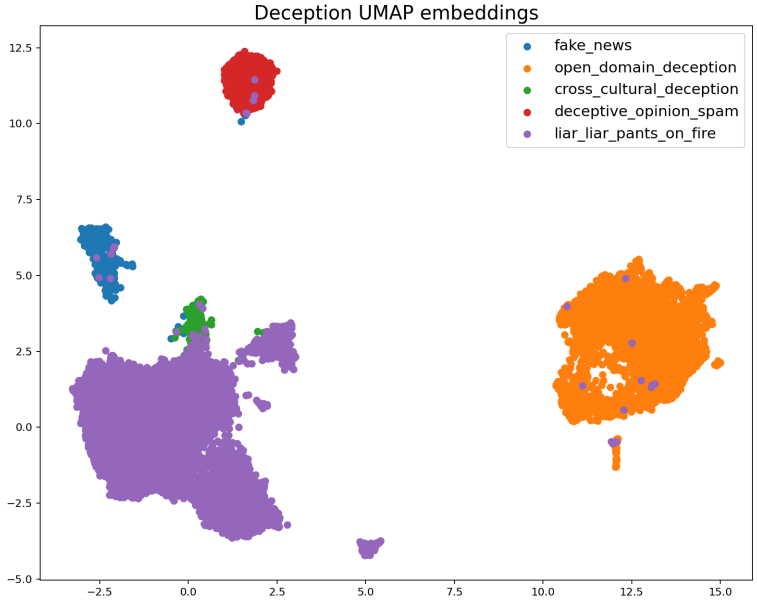

Fig. 2. Deception sentence embeddings using pre-trained BERT

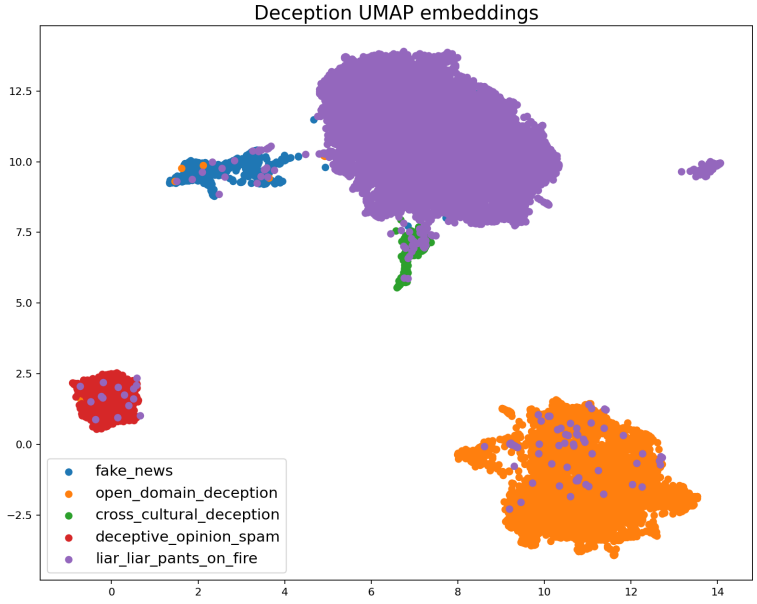

Fig. 3. Deception sentence embeddings using pre-trained DistilBERT

Finally, we previously observed that multi-source training, i.e. training a classification model by combining data from multiple source domains, is not helpful for cross-domain deception classification. We hypothesized that there are conflicting cues to deception in some pairs of domains, which

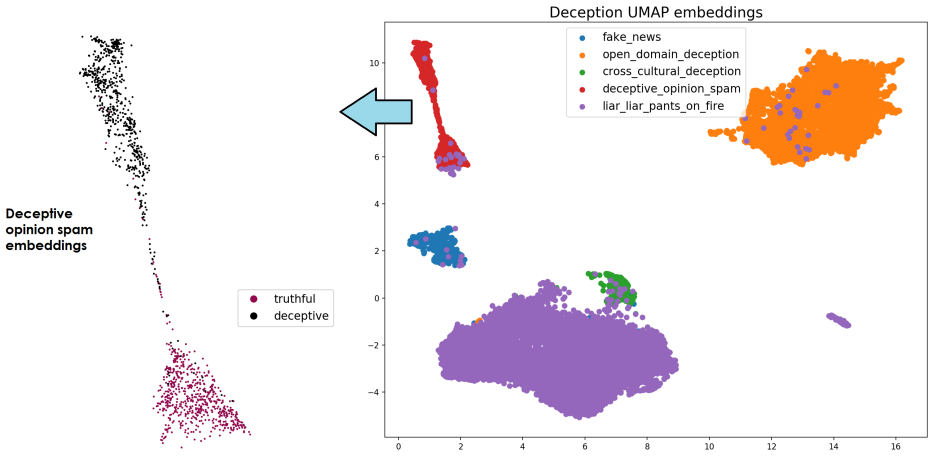

Fig. 4. Deception sentence embeddings after fine-tuning BERT on deceptive opinion spam dataset for deception classification. The zoomed-in view of the deceptive opinion spam embeddings are shown on the left, with deceptive and truthful samples marked with different colours.

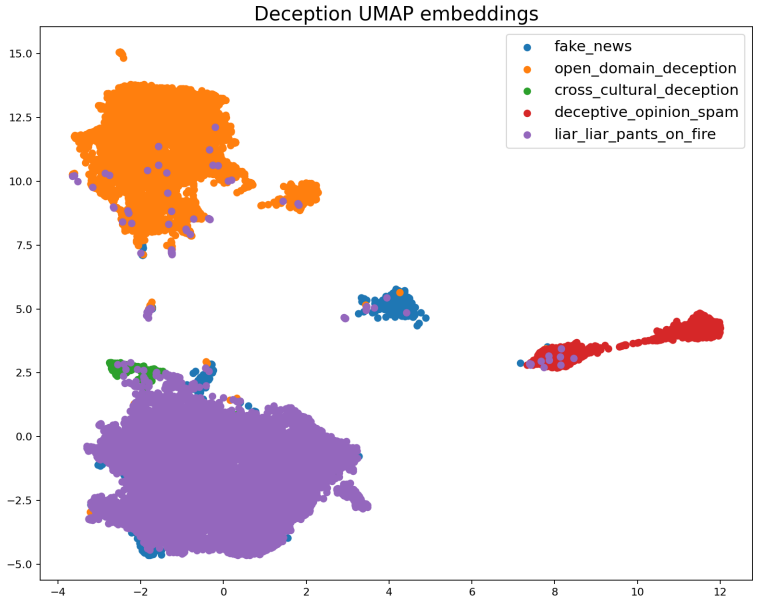

Fig. 5. Deception sentence embeddings after fine-tuning DistilBERT on deceptive opinion spam dataset for deception classification.

may limit the multi-source model's ability to learn useful cues to deception for a particular target domain.

In this section, we aim to gain a deeper understanding of the classification results presented in the previous section. We do this by analyzing the BERT-based models and the text representations that they learn during training. The success of BERT-based models in a wide range of NLP applications

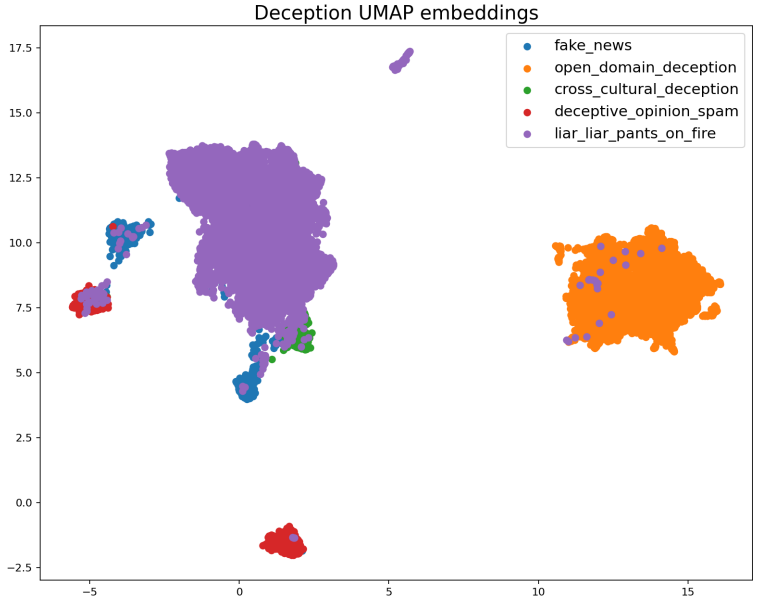

Fig. 6. Deception sentence embeddings after fine-tuning BERT on concatenation of deceptive opinion spam and fake news datasets for deception classification.

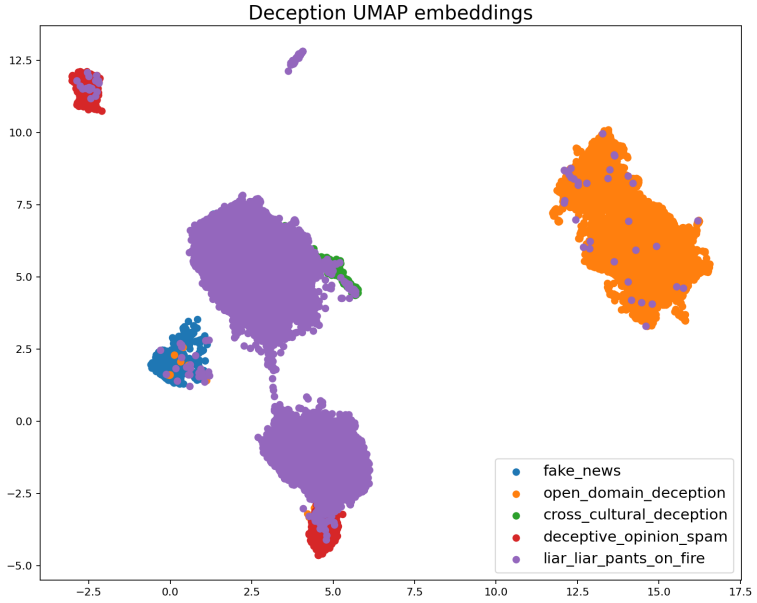

Fig. 7. Deception sentence embeddings after fine-tuning BERT on concatenation of liar liar pants on fire and deceptive opinion spam datasets for deception classification.

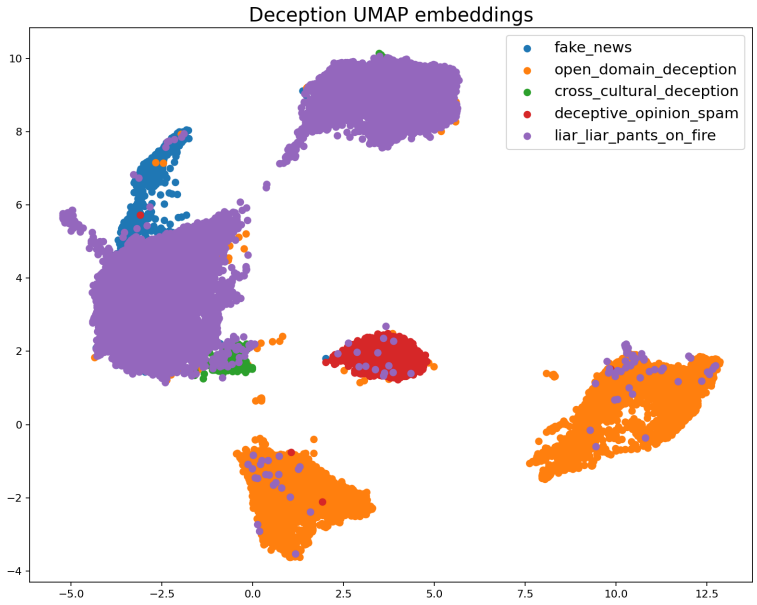

Fig. 8. Deception sentence embeddings after fine-tuning BERT on concatenation of open domain deception and liar liar pants on fire datasets for deception classification.

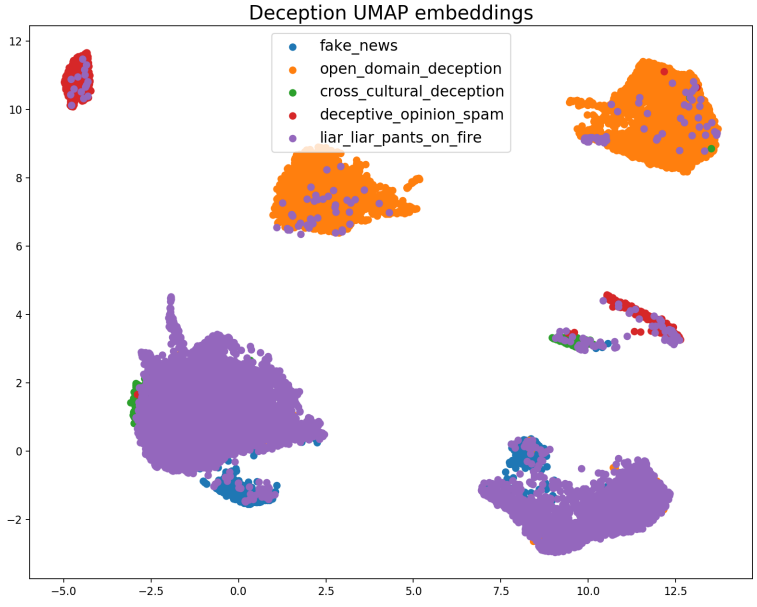

Fig. 9. Deception sentence embeddings after fine-tuning BERT on datasets from all deception domains for deception classification.

has been attributed to the contexualized embeddings that they produce, which are non-static and can serve as a sentence level representation. By visualizing and analyzing the contextualized sentence embeddings from the BERT models, we hope to draw insights about the differences in domains, the effects of fine-tuning, and the limitations of multi-source training.

As in the previous section, we use two variants of BERT: the BERT base model by Devlin et al. [3] and DistilBERT model by Sanh et al. [30]. For both these models, we take the `[CLS]` token's representation to extract sentence level embedding of each sentence. This results in a 768 dimensional vector for each sentence. To visualize the deception sentence embeddings, we project the sentence embeddings into a 2D space using UMAP [18]. Figures 2-9 show the visualization of the sentence embeddings reduced to 2 dimensions, represented by the x and y axes.

5.1 Differences between domains

We begin by studying the embeddings for the 5 deception domains to gain insights about the differences between the domains. Figure 2 shows the deception sentence embeddings visualization for different domains obtained using pre-trained BERT. We observe from this visualization that there are well-defined clusters of embeddings for most domains, for example deceptive opinion spam, in red). In contrast, the liar liar pants on fire dataset, shown in purple, appears to have more broad and diverse embeddings, with several purple data points appearing in each of the other clusters.

Next, we contrast the BERT embeddings with DistilBERT embeddings. The deception sentence embeddings using pre-trained DistilBERT are shown in Figure 3. We notice that the choice of the BERT model does not seem to have a significant effect on the sentence representations, as the visualizations of the sentence embeddings for both BERT and DistilBERT are quite similar.

Both visualizations show that some domains are situated closer to each other in the vector space, while others are more distant from each other. For example, the purple cluster, representing liar liar pants on fire, appears close in proximity to the green cluster, representing cross-cultural deception, as well as to the blue cluster, representing fake news. Notably both the fake news and the liar liar pants on fire domains contain topics related to politics and thus are similar. In contrast, sentence embeddings from open-domain deception (in orange) and deceptive opinion spam (in red) are not close to the other domains in the vector space. Based on these visualizations, we hypothesize that domains that are closer to each other in the vector space are better suited for cross-domain deception detection than those that are more distant to each other. Informally, we observe that the fake news domain achieves the highest performance on the liar liar pants on fire domain in the zero shot setup, compared to the other datasets. We test this hypothesis quantitatively in Section 7.

5.2 Impact of fine-tuning on sentence embeddings

So far we visualized the sentence embeddings extracted from pre-trained BERT based models. Next, to understand the impact of fine-tuning on sentence embeddings, we first fine-tune the BERT model for deception classification. We select one domain for fine-tuning, and then visualize the sentence embeddings after fine-tuning. As an example, we show the sentence embeddings after fine-tuning BERT on the deceptive opinion spam dataset in Figure 4. We note that we explored fine-tuning for each of the 5 deception domains and observed similar trends for all domains, but highlight one domain here for illustrative purposes.

Figure 4 shows the sentences for the training domain – deceptive opinion spam – in red, as well as the sentences for all the other domains. We compare this visualization with the previously visualized sentence embeddings before fine-tuning, which are shown in Figure 2. We observe that after fine-tuning the structure of the sentence embeddings in the deceptive opinion spam domain changes from being a concentrated cluster to two nearly distinct clusters as shown in the zoomed-in

view on the left in Figure 4. Upon manually examine at the samples in the two distinct clusters for deceptive opinion spam, we find that remarkably, each cluster is representative of a given class (truthful/deceptive) and most samples in a cluster have the same class. Specifically the ratio of truthful/deceptive samples per subcluster for deceptive opinion spam is nearly 80:20 in one subcluster and nearly 20:80 in the other. For the domains other than deceptive opinion spam, the sentence embeddings are also adjusted after fine-tuning but there is no clear trend. Also, we note that although fine-tuning separates the sentences in the two classes, it does not necessarily bring the embeddings in different domains closer.

Next, we run the same analysis but this time we fine-tune the DistilBERT model instead of BERT. The resulting sentence embeddings for all the domains including deceptive opinion spam are shown in Figure 9. We observe similar results as we did for the BERT model. We also see a separation between the truthful and deceptive classes for deceptive opinion spam sentences after fine-tuning (shown in red). The structure of the sentence embeddings for the other domains do not change considerably. Overall, BERT and DistilBERT seem to behave similarly when fine-tuned for deception classification. It appears that fine-tuning BERT-based embeddings for a particular deception domain results in a separation of embeddings for that domain into a deceptive cluster and a truthful cluster. However, this separation of the embeddings only occurs for the domain that the model is fine-tuned on; it does not appear to affect the other domain embeddings.

Next, we study the impact on sentence embeddings when more than one deception domain is used for fine-tuning BERT. We observed from our classification experiments that multi-source training was not effective for cross-domain deception detection, and often performed worse than models trained on a single source domain. In this section we aim to understand this somewhat counterintuitive finding by visualizing the embeddings of models trained on multiple domains.

Figure 6 shows a visualization of the sentence embeddings after being fine-tuned on two domains: deceptive opinion spam (in red) and fake news (in blue). We observe that after fine-tuning, the sentence embeddings of these two domains are split into two groups each. Upon manual examination as mentioned in the previous subsection, we again find that the partition is based on the truthful and deceptive samples in each dataset, with around 80% of samples in a subcluster being from the same class.

Similarly, Figure 7 shows the visualized sentence embeddings after fine-tuning on the concatenation of liar liar pants on fire and deception opinion spam datasets for fine-tuning BERT. Again we observe the partitioning of those two domains into two distinct clusters. Yet another such example is shown in Figure 8 where the BERT classifier was fine-tuned on the concatenation of the open domain deception and liar liar pants on fire datasets.

In our final analysis, we fine-tune the BERT classifier on the combination of all the deception domains. Although this is an unrealistic scenario if there is no training data available for a particular target domain, we run this experiment to understand the behaviour of sentence embeddings when datasets from several domains are used to fine-tune the model for deception classification. The sentence embeddings after fine-tuning BERT are shown in Figure 9. As we would expect and in agreement with previous findings, we observe that the fine-tuning separates each domain into two clusters based on the classes (truthful and deceptive). This analysis sheds light on why multi-source training may not be effective for cross-domain classification. It appears that fine-tuning the BERT model on a domain or set of domains effectively partitions the domain embeddings into truthful and deceptive clusters. These clusters are specific to each domain, and may not be useful for transferring knowledge about patterns of deception vs. truth to other domains. The concept of deception in one domain might be different from that in another domain.

Motivated by our analysis of embeddings visualizations, we next aim to quantitatively test our intuitions in the coming sections.

6 DISTANCE BETWEEN DECEPTION DOMAINS

Having observed that there are consistent cross-domain performance gaps, with some more extreme than others, we aim to understand and quantify these differences. We also observed in our visualizations of the learned embeddings that some domains seem to be closer to each other in the vector space, while others are further apart. To better understand and quantify these observations, we define five distance metrics which can be used to measure the distance between a pair of domains. We first formulate the general notion of distance. Let D_S and D_T be the source and target domains respectively. We denote the distance from D_S to D_T as $distance(D_S, D_T)$. We hypothesize that $distance(D_S, D_T)$ is negatively correlated with the cross-domain accuracy, that is, the accuracy that is obtained by training a model on D_S and testing on D_T. Below, we detail the five distance metrics used in this work.

6.1 Vocabulary intersection

The distance between two domains can be defined based on the vocabulary intersection between them. The lower the overlap between the vocabularies of D_S and D_T, the greater the distance between them.

$$distance(D_S, D_T) = 1 - \frac{|V_S \cap V_T|}{|V_S \cup V_T|}, \tag{1}$$

where V_S and V_T are vocabularies of D_S and D_T respectively. The vocabulary intersection based distance is symmetric, that is, $distance(D_S, D_T) = distance(D_T, D_S)$.

6.2 Word frequency distribution

Zipf's law states that the rank of a word on a frequency list multiplied by its frequency is a constant [28]. Although the law holds only approximately [31], we expect it to hold as such across different domains of deception. Drawing inspiration from Barnes et al. [1], we compute the frequency distributions of the top $k = 1,000$ frequent words in D_S and D_T, normalize the frequencies to obtain P and Q respectively, and compute the KL divergence $D_{KL}(P||Q) = \sum_i P(i) log\Big(\frac{P(i)}{Q(i)}\Big)$. The distance between D_S and D_T can be be defined as

$$distance(D_S, D_T) = 1 - \exp\Big(-D_{KL}(P||Q)\Big), \tag{2}$$

where the exponential is used to bring the range to $[0, 1]$. Because of the property that KL divergence is asymmetric, $distance(D_S, D_T) \neq distance(D_S, D_T)$ in general.

6.3 Logistic regression word weights

Let $V = V_S \cup V_T$ be the combined vocabulary of the source and target domains. Using all words in V as features, logistic regression models can be trained to obtain classifiers C_S and C_T respectively for D_S and D_T. Then the distance between D_S and D_T can be computed as

$$distance(D_S, D_T) = \frac{1}{|V|} \sum_{w \in V} |w_S - w_T|, \tag{3}$$

where w_S and w_T are the weights of word $w \in V$ obtained using logistic regression classifiers C_S and C_T, respectively. The distance computed using logistic regression word weights is symmetric, that is, $distance(D_S, D_T) = distance(D_T, D_S)$. We note that for computing distance using logistic regression word weights, we need the gold labels of the samples of both the source and target domains. This requirement is hard to meet in most cases.

6.4 Average sentence embeddings

A sentence can be represented in a vector space using a pretrained sentence embedder such as BERT [3] by using the representation of the [CLS] token. A *domain embedding* can be obtained for a domain by taking the mean of all the sentence representations in that domain. The distance between D_S and D_T can then be computed as the cosine distance between the corresponding domain embeddings.

$$distance(D_S, D_T) = \frac{1 - \cos(SD_S, SD_T)}{2}, \tag{4}$$

where SD_S is the mean of all the sentence embeddings in D_S and SD_T is the mean of all the sentence embeddings in D_T. The distance computed using average sentence embeddings technique is symmetric, that is, $distance(D_S, D_T) = distance(D_T, D_S)$. In our experiments, we extract the sentence embeddings from two BERT based models by freezing or unfreezing the pre-trained weights.

BERT freeze. We take the pre-trained BERT base model from Devlin et al. [3] and for each sentence compute the [CLS] token's representation. This representation is considered as the sentence embedding for a given sentence.

BERT unfreeze. We take the pre-trained BERT base model from Devlin et al. [3]. We add a binary classification head on top and fine-tune it on the task of deception classification using the concatenation of training data from all the five deception domains. We use the concatenation of development data from all the five domains to decide early stopping, by halting training when the development accuracy does not improve for 5 consecutive epochs. For each sentence we compute the [CLS] token's representation (the representation obtained before the classification head). This representation is considered as the sentence embedding for a given sentence.

DistilBERT freeze. This is similar to BERT freeze with the difference that DistilBERT model [30] is used instead of the BERT base model.

DistilBERT unfreeze. This is similar to BERT unfreeze with the difference that DistilBERT model [30] is used instead of the BERT base model.

6.5 Word embeddings similarity distribution

Although BERT based embeddings can be obtained on a sentence level, these embeddings are general and do not explicitly take into account domain specificity in deception. We therefore design a metric which can capture custom deception domains at the word level. Domain-specific word2vec embeddings can be trained using the tokenized sentences in D_S and D_T separately to obtain E_S and E_T respectively. Let W_S be the top $k = 1,000$ frequent words in D_S and W_T be the top k frequent words in D_T. We compute the cosine similarities between all the word embeddings of W_S and all the word embeddings of W_T. For the n-th word ws_n in W_S, we compute the cosine similarity distribution with W_T as a histogram, normalize it and denote it as P_n. Similarly for the n-th word wt_n in W_T, we compute the corresponding normalized histogram Q_n. Then the KL divergence between P_n and Q_n can be computed as

$$D_{KL}(P_n||Q_n) = \sum_i P(i) log\Big(\frac{P_n(i)}{Q_n(i)}\Big) \tag{5}$$

The distance between D_S and D_T can then be quantified as

$$distance(D_S, D_T) = 1 - \exp\left(\frac{-\sum_{j=1}^{k} D_{KL}(P_j||Q_j)}{k}\right) \tag{6}$$

Distance metric	Labels?	Symmetric?
Vocab. intersection	No	Yes
Word freq. dist.	No	No
Log. reg. weights	Yes	Yes
Avg. sentence embed.	No	Yes
Word embed. sim. dist.	No	No

Table 9. Properties of various distance metrics.

This distance metric is asymmetric because $D_{KL}(P_n||Q_n) \neq D_{KL}(Q_n||P_n)$ and therefore, $distance(D_S, D_T) \neq distance(D_T, D_S)$ generally.

For computing the distance between domains, We set $k = 1000$ for both the word frequency distribution based distance and word embeddings similarity distribution based distance.

We summarize the distance metrics and their properties in Table 9. After defining these distance metrics, we study their relationship with cross-domain deception detection in the following section.

7 EVALUATION OF DISTANCE METRICS

To test the usefulness of the various distance measures in capturing information related to cross-domain deception detection, we evaluate the distance metrics using two approaches: (1) Triple-based evaluation, and (2) correlation-based evaluation. Both methods and results are described below.

7.1 Triple-based evaluation

We first evaluate the distance metrics using a triple-based evaluation, to determine whether the distance measures are useful for classifying the domain of a text. We formulate an evaluation task as follows: Given three sentences a, b, c, where sentence a and c are from different domains and b is from the same domain as a, determine whether b shares the domain with a or with c. The domain determination is done using one of the distance measures, and ideally, we would expect a valid distance metric to produce a lower distance between a and b, as compared to a and c.

To create the triples for such evaluation, we first randomly pick 5 sentences from each domain. We create ${}^5C_2$ (a, b) pairs from each domain of the 5 deception domains. For each (a, b) pair, we take the 5 sentences from the rest of the 4 domains. This creates in total $10 \times 5 \times 4 = 1000$ (a, b, c) triples for evaluation.

Here we want to compute the distance between two sentences as opposed to the distance between two domains. Therefore, we adapt the average sentence embeddings based distance metric and the word embeddings similarity distribution based distance metric described earlier to work on a sentence level.

Sentence embeddings cosine distance. Let s_1 and s_2 be two sentences. We compute the distance between them as the cosine distance between the sentence embeddings of s_1 and s_2.

GloVe embeddings similarity distribution. Let two sentences be $S_a = \{w_{a1}, w_{a2}, \ldots, w_{aA}\}$ and $S_b = \{w_{b1}, w_{b2}, \ldots, w_{bB}\}$. We first compute the cosine similarity of $Emb(w_{ak})$ with $Emb(w_{aj})$ for $j \neq k$ and $\forall k \in [1, A]$ and $j \in [1, A]$, where embedding of a word is computed using pretrained GloVe 100-dimensional embeddings [21]. Then we compute the cosine similarity histogram H_a using the above similarity (the histogram has similarity on the x-axis and frequency on the y axis). We repeat the above two steps for sentence S_b to obtain H_b. Finally the distance between S_a and S_b is computed as the KL divergence between H_a and H_b.

We evaluate how many times out of 1000 triples the distance metric correctly scores the distance between a and b to be less than that between a and c. Since randomization is involved while picking the 5 sentences from each domain, We run the evaluation for each of the two distance metrics 10 times and show the accuracy in Table 10. As shown in the table, the sentence embeddings cosine distance achieves an accuracy of 81%, suggesting that this distance metric captures useful information about the distance between deception domains.

Distance metric	Accuracy
Sentence embeddings cosine distance	0.81 ± 0.05
GloVe embeddings similarity distribution	0.63 ± 0.04

Table 10. Evaluation of distance metrics using triples.

7.2 Correlation with cross-domain accuracy

We further hypothesize that the distance between D_S and D_T is negatively correlated with the cross-domain accuracy. That is, as the distance between a source and target domain increases, the classification performance between the domains decreases. To test this hypothesis, we computed Pearson correlations between the distances and the respective cross-domain accuracies computed using our deception classification model. We focus on the BERT unfrozen model which had obtained the best cross-domain performance overall. For each distance metric, we compute 4 distance-accuracy pairs for a given source domain by taking into account all possible cross-domain combinations. We then combine the distance-accuracy pairs across all the source domains to obtain 20 (4 target domains for each of the 5 source domains) distance-accuracy pairs. We compute the Pearson correlations for each distance metric and report the correlations in Table 11. We observe that all the distance metrics have a negative correlation coefficient, and this correlation was statistically significant ($p < 0.05$) for average sentence embeddings using DistilBERT frozen embeddings. The average sentence embeddings and the word frequency distribution based distance measures showed comparatively stronger negative correlations of -0.519 and -0.357 respectively. Note that the BERT vs DistilBERT results differ while freezing and fine-tuning the pre-trained weights. However, the earlier results which just focused on accuracies only (see Table 5) indicated that BERT and DistilBERT show similar trends while freezing and fine-tuning the pre-trained weights. The strong negative correlation suggests that the distance between domains such as the average sentence embeddings is a useful metric for determining whether a deception model from a source domain can be reliably applied to a target domain. Motivated by this finding, we aim to improve cross-domain deception detection by leveraging domain distance information. We explore this in the following section.

8 RECOMMENDATION USING CROSS-DOMAIN DISTANCE MEASURES

Our classification results in Section 4 have shown that there are substantial differences in cross-domain deception classification of a target domain domain, depending on which source domain is used. Further, the results show that simply combining multiple domains and training a multi-source classification model often performs worse than using a single domain. Our analysis of the embeddings learned by the model via visualization provided further insights into these findings, and led us to develop measures of domain distance to quantify the intuitions that we gained from visualization the embeddings. In our evaluation of those distance metrics, we found that some measures of distance are negatively correlated with cross-domain classification performance,

Distance metric	Correlation coefficient
(1) Vocab. intersection	-0.253
(2) Word freq. dist.	-0.357
(3) Log. reg. weights	-0.269
(4) Avg. sentence embed. [bert frozen]	-0.344
(4) Avg. sentence embed. [distilbert frozen]	**-0.519***
(4) Avg. sentence embed. [bert finetuned]	-0.137
(4) Avg. sentence embed. [distilbert finetuned]	-0.092
(5) Word embed. sim. dist.	0.095

Table 11. Pearson correlations between cross-domain distance and cross-domain accuracy for different distance metrics. The significant correlations are denoted by a *.

suggesting that domains that are distant from each other in the vector space perform poorly at cross-domain classification when paired as source-target domains.

All of these findings motivate the experiments presented in this section. We aim to develop a classification approach that leverages the notion of domain distance to improve cross-domain deception detection. The main idea is as follows: given a target domain, find the optimal source domain to use for training a deception detection model. To do this, we use the cross-domain distance measures that we previously computed in Section 6. We compute the domain distance between the target domain and all possible source domains. Then, we recommend the source domain which has the smallest distance from the target domain.

We compare the performance of this recommender system with 2 baselines: (1) A random recommendation system which chooses a source domain uniformly at random for a given target domain. To get a reliable cross-domain accuracy, we consider 100000 trials of random recommendation and calculate the average cross-domain accuracy across all trials. (2) Multi-source leave-one-out training, which combines all source domains, excluding the target domain, for classification. The recommendation results are shown in Table 12. The table shows both the accuracy and rank of the accuracy upon using the recommended source domain for a given target domain. It compares the results obtained via different distance metrics, and also compares these distance-based recommendations with the results of the 2 baseline models.

We observe that the recommendations obtained using sentence embeddings based distance metrics are the most useful among other distance metrics. This complements the correlation results from the previous section, which showed that the sentence embeddings based distance metrics were most negatively correlated with cross-domain classification. The best performance, averaged across all target domains, was obtained using the BERT frozen sentence embeddings distance. That BERT frozen distance-based recommender achieved an average accuracy of 55.04%, and outperformed other distance-based recommenders as well as the 2 baseline systems. In terms of the rank of recommendation, we observe that the average rank in this case is 1.8.

Additionally, we find that while recommending a source domain is a relatively easier task for some target domains, recommendation is difficult in some other domains. For example, for the target domains fake news and open domain deception, most distance metrics get the recommendation right in a majority of cases. However, this is more challenging for liar liar pants on fire as the target domain, since no model achieves an accuracy that is significantly above 50%. We also observe that the recommendation using distance metrics is better than both random recommendation and leave one out multisource recommendation. This is an important use case of distance metrics, showing that they can reliably be used for improving cross-domain performance.

Distance metric	Target domain											
	FN		ODD		CCD		DOS		LLPF		Average	
	Acc	Rank	Acc	Rank	Acc	Rank	Acc	Rank	Acc	Rank	Acc	Rank
(1) Vocab. intersection	**0.62**	1	0.478	2	0.504	2	0.52	2	0.5	4	0.5244	2.2
(2) Word freq. dist.	0.572	2	0.581	1	0.456	4	0.520	2	0.500	4	0.526	2.6
(3) Log. reg. weights	0.500	4	0.474	3	**0.566**	1	0.520	2	0.500	4	0.512	2.8
(4) Avg. sent. embed. [bert frozen]	**0.620**	1	**0.581**	1	0.501	3	**0.550**	1	0.500	3	**0.550**	**1.8**
(4) Avg. sent. embed. [distilbert frozen]	**0.620**	1	**0.581**	1	0.501	3	0.520	2	0.500	4	0.544	2.2
(4) Avg. sent. embed. [bert finetuned]	**0.620**	1	**0.581**	1	0.501	3	0.453	4	0.500	4	0.531	2.6
(4) Avg. sent. embed. [distilbert finetuned]	**0.620**	1	**0.581**	1	0.504	2	0.453	4	**0.504**	2	0.532	2
(5) Word embed. sim. dist.	**0.620**	1	0.4	4	0.504	2	0.453	4	0.500	4	0.495	3
Multisource leave one out	0.541	-	0.500	-	0.550	-	0.447	-	0.521	-	0.512	-
Random recommendation	0.553	2.5	0.484	2.5	0.507	2.5	0.506	2.5	0.503	2.5	0.511	2.5
Best possible recommendation	0.620	1	0.581	1	0.566	1	0.550	1	0.506	1	0.565	1

Table 12. Cross-domain accuracies and ranks upon recommending using various distance metrics for different target domains. (**FN**: Fake news, **ODD**: Open domain deception, **CCD**: Cross-cultural deception, **DOS**: Deceptive opinion spam, **LLPF**: Liar liar pants on fire)

9 CONCLUSION

In this paper, we conducted experiments to study the generalization ability of deception models to new domains. We trained baseline models of deception and identified performance gaps between within-domain and cross-domain performance across five domains. Our extensive classification experiments show the relative strengths and weaknesses of various models, and we identified the importance of fine-tuning BERT-based models. We explored the impact of the amount of target training data on classification performance, comparing zero shot, few shot, and full shot learning conditions. In our comparison of multi-source training with single-source training, we found that combining domains for classification generally performs worse than using a single optimal source domain.

To further understand underlying challenges to cross-domain deception detection, we visualized the sentence embeddings extracted from BERT based models. We observed that some domains appeared closer in the vector space than others, and also examined the effects of fine-tuning on the embeddings learned by the models. Based on these visualizations, we hypothesized that domain distance may be related to cross-domain performance. We proposed five distance metrics that can measure the distance between a pair of domains and experimentally showed that the distance between a pair of deception domains is negatively correlated with the cross-domain accuracy. Finally, we developed a method to utilize the distance metrics to recommend source domain that would be optimal when chosen for a given target domain. The recommendation performed better than using a model trained on multiple source domains.

Our results highlight the need to develop robust models of deception detection that can generalize to new domains by taking cues from the distance between the source and target domains. As future work, we will continue to explore methods to improve cross-domain deception detection by potentially minimizing the distance between source and target domains during training. Hopefully, these efforts will lead to continued improvement in deception detection models which will be better equipped to handle diverse examples of real-world deception.

REFERENCES

[1] Jeremy Barnes, Roman Klinger, and Sabine Schulte im Walde. 2018. Projecting Embeddings for Domain Adaption: Joint Modeling of Sentiment Analysis in Diverse Domains. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 818–830. https://www.aclweb.org/anthology/C18-1070

[2] Pasquale Capuozzo, Ivano Lauriola, Carlo Strapparava, Fabio Aiolli, and Giuseppe Sartori. 2020. DecOp: A Multilingual and Multi-domain Corpus For Detecting Deception In Typed Text. In *Proceedings of the 12th Language Resources and Evaluation Conference*. European Language Resources Association, Marseille, France, 1423–1430. https://www.aclweb.

org/anthology/2020.lrec-1.178

[3] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 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)*. Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[4] Eileen Fitzpatrick and Joan Bachenko. 2016. Estimating the amenibility of new domains for deception detection. In *Proceedings of the Second Workshop on Computational Approaches to Deception Detection*. Association for Computational Linguistics, San Diego, California, 26–31. https://doi.org/10.18653/v1/W16-0804

[5] Tommaso Fornaciari, Federico Bianchi, Massimo Poesio, and Dirk Hovy. 2021. BERTective: Language Models and Contextual Information for Deception Detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*. Association for Computational Linguistics, Online, 2699–2708. https://doi.org/10.18653/v1/2021.eacl-main.232

[6] Tommaso Fornaciari, Leticia Cecilia Cagnina, Paolo Rosso, and Massimo Poesio. 2020. Fake opinion detection: how similar are crowdsourced datasets to real data? *Language Resources and Evaluation* (2020), 1–40.

[7] Tommaso Fornaciari and Massimo Poesio. 2013. Automatic deception detection in Italian court cases. *Artificial intelligence and law* 21, 3 (2013), 303–340.

[8] Tommaso Fornaciari and Massimo Poesio. 2014. Identifying fake Amazon reviews as learning from crowds. (2014).

[9] Maria Glenski, Ellyn Ayton, Robin Cosbey, Dustin Arendt, and Svitlana Volkova. 2020. Towards Trustworthy Deception Detection: Benchmarking Model Robustness across Domains, Modalities, and Languages. In *Proceedings of the 3rd International Workshop on Rumours and Deception in Social Media (RDSM)*. Association for Computational Linguistics, Barcelona, Spain (Online), 1–13. https://www.aclweb.org/anthology/2020.rdsm-1.1

[10] Ángel Hernández-Castañeda, Hiram Calvo, Alexander Gelbukh, and Jorge J García Flores. 2017. Cross-domain deception detection using support vector networks. *Soft Computing* 21, 3 (2017), 585–595.

[11] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*. https://doi.org/10.5281/zenodo.1212303

[12] Minqing Hu and Bing Liu. 2004. Mining and Summarizing Customer Reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (Seattle, WA, USA) *(KDD '04)*. Association for Computing Machinery, New York, NY, USA, 168–177. https://doi.org/10.1145/1014052.1014073

[13] Swapnil Jain, Shrikant Malviya, Rohit Mishra, and Uma Shanker Tiwary. 2017. Sentiment Analysis: An Empirical Comparative Study of Various Machine Learning Approaches. In *Proceedings of the 14th International Conference on Natural Language Processing (ICON-2017)*. NLP Association of India, Kolkata, India, 112–121. https://www.aclweb.org/anthology/W17-7515

[14] Stefan Kennedy, Niall Walsh, Kirils Sloka, Andrew McCarren, and Jennifer Foster. 2019. Fact or Factitious? Contextualized Opinion Spam Detection. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*. Association for Computational Linguistics, Florence, Italy, 344–350. https://doi.org/10.18653/v1/P19-2048

[15] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A Method for Stochastic Optimization. http://arxiv.org/abs/1412.6980 cite arxiv:1412.6980Comment: Published as a conference paper at the 3rd International Conference for Learning Representations, San Diego, 2015.

[16] Sarah Ita Levitan, Angel Maredia, and Julia Hirschberg. 2018. Linguistic cues to deception and perceived deception in interview dialogues. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*. 1941–1950.

[17] Jinfen Li, Zhihao Ye, and Lu Xiao. 2019. Detection of Propaganda Using Logistic Regression. In *Proceedings of the Second Workshop on Natural Language Processing for Internet Freedom: Censorship, Disinformation, and Propaganda*. Association for Computational Linguistics, Hong Kong, China, 119–124. https://doi.org/10.18653/v1/D19-5017

[18] Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software* 3, 29 (2018), 861.

[19] Rada Mihalcea and Carlo Strapparava. 2009. The Lie Detector: Explorations in the Automatic Recognition of Deceptive Language. In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*. Association for Computational Linguistics, Suntec, Singapore, 309–312. https://www.aclweb.org/anthology/P09-2078

[20] Myle Ott, Yejin Choi, Claire Cardie, and Jeffrey T. Hancock. 2011. Finding Deceptive Opinion Spam by Any Stretch of the Imagination. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, Portland, Oregon, USA, 309–319. https://www.aclweb.org/anthology/P11-1032

[21] Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, 1532–1543. https://doi.org/10.3115/v1/D14-1162

[22] Verónica Pérez-Rosas, Mohamed Abouelenien, Rada Mihalcea, Yao Xiao, CJ Linton, and Mihai Burzo. 2015. Verbal and nonverbal clues for real-life deception detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. 2336–2346.

[23] Verónica Pérez-Rosas, Bennett Kleinberg, Alexandra Lefevre, and Rada Mihalcea. 2018. Automatic Detection of Fake News. In *Proceedings of the 27th International Conference on Computational Linguistics*. Association for Computational Linguistics, Santa Fe, New Mexico, USA, 3391–3401. https://www.aclweb.org/anthology/C18-1287

[24] Verónica Pérez-Rosas and Rada Mihalcea. 2014. Cross-cultural Deception Detection. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics, Baltimore, Maryland, 440–445. https://doi.org/10.3115/v1/P14-2072

[25] Verónica Pérez-Rosas and Rada Mihalcea. 2015. Experiments in Open Domain Deception Detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Lisbon, Portugal, 1120–1125. https://doi.org/10.18653/v1/D15-1133

[26] Denis Peskov, Benny Cheng, Ahmed Elgohary, Joe Barrow, Cristian Danescu-Niculescu-Mizil, and Jordan Boyd-Graber. 2020. It Takes Two to Lie: One to Lie, and One to Listen. (2020).

[27] Martin Potthast, Johannes Kiesel, Kevin Reinartz, Janek Bevendorff, and Benno Stein. 2017. A stylometric inquiry into hyperpartisan and fake news. *arXiv preprint arXiv:1702.05638* (2017).

[28] David M. W. Powers. 1998. Applications and Explanations of Zipf's Law. In *New Methods in Language Processing and Computational Natural Language Learning*. https://www.aclweb.org/anthology/W98-1218

[29] Yafeng Ren and Yue Zhang. 2016. Deceptive Opinion Spam Detection Using Neural Network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, 140–150. https://www.aclweb.org/anthology/C16-1014

[30] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2020. DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter. arXiv:1910.01108 [cs.CL]

[31] Bengt Sigurd, Mats Eeg-Olofsson, and Joost Van Weijer. 2004. Word length, sentence length and frequency–Zipf revisited. *Studia Linguistica* 58, 1 (2004), 37–52.

[32] Felix Soldner, Verónica Pérez-Rosas, and Rada Mihalcea. 2019. Box of lies: Multimodal deception detection in dialogues. 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)*. 1768–1777.

[33] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. Association for Computational Linguistics, Brussels, Belgium, 353–355. https://doi.org/10.18653/v1/W18-5446

[34] William Yang Wang. 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, Vancouver, Canada, 422–426. https://doi.org/10.18653/v1/P17-2067

[35] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-Art Natural Language Processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online, 38–45. https://doi.org/10.18653/v1/2020.emnlp-demos.6

[36] Wenlin Yao, Zeyu Dai, Ruihong Huang, and James Caverlee. 2017. Online Deception Detection Refueled by Real World Data Collection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing, RANLP 2017*. INCOMA Ltd., Varna, Bulgaria, 793–802. https://doi.org/10.26615/978-954-452-049-6_102

[37] Wenjie Yin and Arkaitz Zubiaga. 2021. Towards generalisable hate speech detection: a review on obstacles and solutions. *PeerJ Computer Science* 7 (June 2021), e598. https://doi.org/10.7717/peerj-cs.598