Computational Models of Argument
H. Prakken et al. (Eds.)

doi:10.3233/FAIA200489

Generating Adversarial Examples for Topic-Dependent Argument Classification

Tobias MAYER [a], Santiago MARRO [a], Elena CABRIO [a] and Serena VILLATA [a]
[a] *Université Côte d'Azur, CNRS, Inria, I3S, France* [1]

Abstract. In the last years, several empirical approaches have been proposed to tackle argument mining tasks, e.g., argument classification, relation prediction, argument synthesis. These approaches rely more and more on language models (e.g., BERT) to boost their performance. However, these language models require a lot of training data, and size is often a drawback of the available argument mining data sets. The goal of this paper is to assess the *robustness* of these language models for the argument classification task. More precisely, the aim of the current work is twofold: first, we generate adversarial examples addressing linguistic perturbations in the original sentences, and second, we improve the robustness of argument classification models using adversarial training. Two empirical evaluations are addressed relying on standard datasets for AM tasks, whilst the generated adversarial examples are qualitatively evaluated through a user study. Results prove the robustness of BERT for the argument classification task, yet highlighting that it is not invulnerable to simple linguistic perturbations in the input data.

Keywords. Argument Mining, Argument Classification, Robustness, Adversarial training

1. Introduction

Argument(ation) Mining (AM) [9,2,8] is the research area aiming at extracting and classifying argumentative structures from text. One subtask is topic-dependent argument classification, where the goal is to find relevant arguments for a given topic or claim from heterogeneous sources. This task is currently addressed by employing state-of-the-art deep learning methods, that recently benefit from pre-trained Language Models (LM) like BERT [3]. The idea underlying LM pre-training is to learn a task-independent understanding of natural language in an unsupervised fashion, from vast amounts of unlabeled text. After learning this general knowledge about a language, the model is then fine-tuned on tasks where the amount of available annotated data is significantly smaller, as it holds for AM annotated datasets. However, AM is a very context-dependent task and requires deep Natural Language Understanding (NLU), raising the research question: *How well does the pre-trained NLU scale in fine-tuned models for specific tasks such as argument classification?* In this paper, we answer this question by breaking it down into the following subquestions: *i) How vulnerable are argument classification models to adversarial*

[1] This work is partly funded by the French government labelled PIA program under its IDEX UCA JEDI project (ANR-15-IDEX-0001) and supported through the 3IA Côte d'Azur Investments in the Future project managed by the National Research Agency (ANR) with the reference number ANR-19-P3IA-0002.

attacks? and *ii) Can the robustness of argument classification models be improved with adversarial training?*
To answer these questions, we evaluate the efficiency of simple linguistic attacks against topic-dependent argument classification models based on LM pre-training. We generate eight different types of perturbations ranging from punctuation deletion to various word-based transformations, i.e. substitution or insertion, preserving the semantics of the sentence. The purpose of these attacks is to make the model more robust with adversarial training. The way we evaluate our approach to assess and improve the robustness of argument classification models is twofold: on the one side, we evaluate the success rate of each perturbation type on a model trained without any adversarial examples, and on the other side, we evaluate the improvement in performance on the original test set after augmenting the training data during adversarial training. For our experimental setting, we rely on two standard datasets in argument mining, namely the *UKP Sentential Argument Mining Corpus* [15], and the *IBM Debater: Evidence Sentences* corpus [14].
To summarize, the main contributions of this paper are the following:

- we propose different ways of creating linguistically simple perturbations and evaluate their impact on current state-of-the-art LM-based argument classification models, with respect to both in-domain and cross-topic performance;
- we address a user study to assess the quality of the generated perturbations;
- we empirically evaluate the effect of adversarial training for argument classification.

Obtained results highlight the effectiveness of adversarial training for argument classification. Furthermore, they point out the relatively robustness of LM that are nevertheless not invulnerable to simple changes to the input data. To the best of our knowledge, this is the first approach to generate natural language adversarial example for AM tasks.

In the following, Section 2 presents the related work. In Section 3, we discuss the methodology and background for adversarial attacks in NLP, and then we focus on adversarial training on the argument classification task. We detail our experimental setting[2], including the used datasets and the generated perturbations in Section 4, and we discuss the obtained results in Section 5. Concluding remarks and future work directions end the paper.

2. Related Work

Despite recent breakthroughs in modelling natural language understanding, the employed neural architectures still lack interpretability. They are black boxes for which it is hard to determine what they exactly learn or are receptive for. In this context, it was found that deep neural networks (DNN) are vulnerable to adversarial attacks; small changes to the input which fool the model into predicting a wrong label. Originally, crafting adversarial examples and attacking DNNs stems from the image processing domain [16,4,18]. Most of the employed methods there are gradient-based. These techniques cannot be easily adopted in the natural language processing domain. Images consist of pixels, which are represented as real value vectors: it is possible to slightly change the pixel values in a way which manipulates the gradients in a forward pass of a model to change the

[2] Code available at: `https://gitlab.com/tomaye/comma2020-adversarial_examples`

prediction, while the image is still perceived as unchanged to a human. On the other hand, modifying a sentence in a way that a human will not notice that change is almost impossible. The main problem here is that while pixel values are represented in a continuous space, words - that can also be represented in a continuous space in the form of real value vectors, i.e., embeddings, - per se are in a discrete space. Theoretically, one could find a vector in the embedding space which changes the prediction of a model, but constructing this vector from a discrete space of words is impossible in most of the cases. So, the recommended option is to create a perturbation on a linguistic level in the target sentence. But as said before, adding a word is most likely perceived by a human, contradicting the idea of an unnoticeable difference. Furthermore, adding even a single word might drastically change the semantics of a sentence. Given these two challenges, adversarial examples in the NLP domain need to be carefully designed. Due to the nature of the problem, only limited work on the perceivability has been done so far. The main work focuses on semantic preserving techniques accepting that the perturbation might be noticed by the human eye [20].

A strategy to generate adversarial examples are black-box approaches. Contrary to white-box approaches, they do not need any model specific knowledge except the input and output. Recent black-box approaches comprise methods concatenating, editing or substituting words in the input sentence [20]. There are also approaches which work on changing the underlying syntax by creating paraphrases [6]. We experimented with this automatic paraphrasing technique to generate adversarial examples. While this is a highly interesting topic, for the argument classification datasets the produced paraphrases were ungrammatical most of the time. So, we decided not to further pursue this kind of perturbation and exclude them from our experiments. An intuitive way of creating perturbations is to replace words with semantically similar alternatives, e.g., synonyms. Alzantot et al. [1] employ an approach where they replace each word of a sentence until the prediction changes. We do apply the same technique of replacing words with semantically similar alternatives, but with a different strategy: we only replace one word at a time minimizing the risk of producing a meaningless sentence. Moreover, we also add adverbs which change the semantics, strictly speaking, but do not change the label from argumentative to non-argumentative. Concerning the model we are attacking, previous work has shown that self-attentive models are more robust than recurrent architectures [5]. While in this work the authors used a white-box approach to precisely aim at weak points of the self-attending model, we went for a model independent black-box strategy. The generated adversarial examples lay the foundation to evaluate the robustness of argument classification models and to improve it with adversarial training.

3. Preliminaries

In this section, we introduce the terminology and give an overview of the methodology for adversarial attacks on deep neural networks (DNN) for NLP. We closely follow the definitions given in [20,18] and explain which setting we chose for the topic-dependent argument classification task.

Perturbation: A perturbation is a minor change to the test input example for the DNN. The goal is to change the prediction of the model, while the modification of the input example should not be perceived by humans. As previously mentioned, the notion of being imperceptible by humans is not as easily applicable to text, because most of the time a change in characters or even words is more obvious to human judgment than a slight adjustment to pixel values. Thus, for NLP the point of perceivability is rather interpreted as preserving the semantics of the original sentence with being still grammatical as a further constraint. Both of these constraints are challenging NLP tasks by themselves and have not been fully solved so far. As a consequence, automatically generated perturbations might violate these constraints raising the necessity for a human evaluation of the generated perturbations.

Granularity of Perturbation: The notion of granularity follows the thought above. While slight changes in single characters might not be that perceivable and preserve semantics as well as syntax, deleting, inserting or replacing words is a different level of perturbation. Even changes on sentence level are possible, e.g. paraphrasing or even adding whole sentences as it was done for attacking reading comprehension models [7]. For the argument classification task, the majority of our perturbations are on word level, since we wanted to evaluate the robustness of the targeted DNN language model against comparatively simple linguistic attacks.

Adversarial Example: An adversarial example x' is a perturbation of an input example x, where the modification indeed changes the prediction Y of the model, so that $y' \neq y$.

Attack Target: An adversarial attack can be targeted to change only specific labels in a multi-class classification setting. For argument classification, we do not see the necessity to specifically target the attacks against a certain label for two reasons: first, argument classification is usually limited to a two or three class classification problem, and second we do not want to make any assumptions about the architecture of the model we are attacking, leading us to the next point.

Model Knowledge: There are different strategies to generate adversarial examples depending on the availability of knowledge about the DNN the attacks are aimed at. White-box approach have access to all the information of the model, e.g. architecture, (hyper-) parameters, loss and activation function, training data, or confidence scores. On the contrary, the black-box approaches have only access to the input and output of a model [11]. We selected a specific model to attack, i.e. BERT, but since there are and will be other self-attending architectures based on language model pre-training, we do not want our perturbations to be limited to only BERT and decided to go for a black-box approach ignoring valuable information like the attention scores.

Adversarial Training: Currently, the only defense strategy against adversarial attacks is adversarial training where the DNN is re-trained with adversarial examples [20,16]. One strategy is also to include inputs which are unlikely to occur naturally. This defense strategy aims at reducing the *"fundamental blind spots"* [4] of a model making the model more robust against divers input. With respect to NLP and specifically to argument classification, this means that including ungrammatical examples in training the model is justified. After all, argument classification is based on representations of full sentences, which are created from word level representations independent of the grammaticality of the sentence.

Evaluation Metric: The evaluation of adversarial attacks can be measured by the degree it decreases the performance of a DNN. We decided to not do that, because we cannot ensure the same number of generated perturbations per input example and thus might bias the results. Another prominent way to evaluate the perturbation efficiency is the success rate. This is the percentage of adversarial examples over the number of generated perturbations.

Robustness: In our terminology, robustness refers to the ability of a model to correctly classify unseen test data from the same domain as the training data. Contrary to that, we refer to generalizability as the concept of being able to exploit the already acquired knowledge in a new domain. For argument classification, this means that when training and test set talk about the same topics, e.g. *abortion*, adversarial attacks are testing robustness. For the case when the test set contains topics which are never seen during training, we talk about (cross-topic) generalizability of a model. Our main goal with adversarial training is to increase the robustness of a model, not its generalizability.

4. Experiment Setup

This section describes *i)* the datasets used for training and testing and the attacked DNN, *ii)* the different types of generated perturbations, and *iii)* a qualitative evaluation of the perturbations through a user study.

4.1. Data and Target Model

As previously mentioned, the application domain for the adversarial attacks in our work is topic-dependent argument classification. For this task, there are two major corpora available: 1) The *UKP Sentential Argument Mining Corpus* [15], which is a collection of 25,492 sentences annotated as an *ArgumentFor (**Arg+**), ArgumentAgainst (**Arg-**)* or *NoArgument (**NoArg**)* to a specific topic. The corpus comprises 8 different topics, i.e. *abortion, cloning, death penalty, gun control, marijuana legalization, minimum wage, nuclear energy* and *school uniforms*, and 2) the *IBM Debater: Evidence Sentences* [14], which is a collection of sentences from online debate portals annotated with *evidence **(Arg)*** or *no evidence **(NoArg)*** in regard to one of the 118 topics. Following existing experimental setups from the literature [14,13], the training set comprises 83 topics (4,065 sentences) and the test set 35 (1,718 sentences).

Self-attentive transformer models like BERT [3], which use LM pre-training, have become a mighty tool for many NLP tasks. This also applies to argument mining. Following recent state-of-the-art on topic-dependent argument classification [13], we evaluate the adversarial attacks on the BERT base model. The input for BERT consists of the input sentence concatenated with the topic. As introduced before, our perturbations are black-box methods not taking advantage of model specific knowledge, e.g. attention score. Thus, they can be easily transferred to other architectures in the future.

We conducted two lines of experiments. The first one to test the success rate of the perturbations, and the second one to evaluate adversarial training. For both lines, training and performance evaluation were based on the code provided by Reimers et al. [13]. Hyper-parameters for fine-tuning the models were also replicated without any changes. The only difference is that we do not split the training data into a development set, since we are not tuning any parameters. For both lines of experiments, there are three different scenarios: 1) a model were the train (80%) and test (20%) sets comprise all eight topics of the UKP corpus (**UKP all**); 2) the leave-one-out training (**UKP x-topic**), where seven topics of the UKP corpus were used for training and the eighth is used for testing. In total, this results in eight different models. The results in this scenario are reported as the average over the eight models; 3) in the last scenario, a model is trained on the IBM corpus with the train-test split described above (**IBM x-topic**).

For the first line of experiments, i.e., perturbation evaluation, the success rate of a perturbation is evaluated on a model trained without any adversarial examples. Only perturbations from the test set are considered in calculating the success rate. For each perturbation, we computed a label-wise success rate. For the second line of experiments, i.e., adversarial training, only perturbations of the training set are considered for augmenting the training data. We re-trained every model under the same conditions as before, but with the only difference being the augmented training data. The evaluation of an adversarially trained model is done on the same unmodified test set as the normally trained counterpart to guarantee comparability.

4.2. Perturbation Types

In the following, we introduce the eight different methods we used to generate perturbations for given input examples. The perturbation generation methods are based on word or token types. Hence, the number of generated perturbations per input example varies. To give an idea of the order of magnitude, we report the average number of generated perturbations for each test set of the two corpora.

Named Entities (NE) The first method we propose consists of replacing a named entity in the input sentence. To achieve this, we constructed a list of named entities for each of the four standard categories, i.e., *PER*, *LOC*, *ORG*, *MISC*, present in the CoNLL 2003 Shared Task dataset for named entity recognition [17]. Using this list, we then generate for each NE present in the original sentence one new perturbation replacing the entity with a different entity from the same category. In order to preserve the semantics, we used pre-trained word embeddings (fastText) as a means of distance, and selected the closest neighbours. If the original input sentence does not contain a NE, no perturbations are generated. Accordingly, the average number of generated perturbations per input sentence varies. On the UKP dataset we produced an average of 3.11 perturbations per sentence. The IBM dataset contains more NEs per sentence, therefore the produced number of perturbations per example is higher, namely 10.15.

Example 4.1 *Original sentence: According to **FBI** statistics, 46,313 Americans were murdered with firearms during the time period of 2007 to 2011.*
*Adversarial attack: According to **U.S. Bureau of Investigation** statistics, 46,313 Americans were murdered with firearms during the time period of 2007 to 2011.*

Adjectives This method is similar to the list-based attack proposed in [1], where words in the input sentence are replaced with a word from a list of semantically similar words. Contrary to the aforementioned work, we only replace one word per perturbation. Specifically, we exchange adjectives with their synonyms, e.g. *big* with *large*, producing one perturbation example for each adjective in the sentence. The synonyms were taken from the WordNet interface in the NLTK. For the UKP dataset, we have an average of 2.12 adjectives per sentence, while for the IBM dataset we generate 2.9 perturbations per sentence.

Punctuation This is the only modification of a sentence on character-level. Here, all the punctuation, e.g., "." or ",", is removed from the original input sentence. Naturally, this method provides one perturbation per sentence.

Scalar Adverbs This method is about adding or replacing emphasising modal adverbs, such as *considerably*, or trigger words for scalar implicature, such as *comparatively* or *largely*. They are added before a verb or an adjective. As will be shown in succeeding sections, the positioning algorithm needs to be improved, since some adverbs should be placed only after the word, while others should be placed only before the word or can take both positions. The average amount of perturbations generated per input sentences is around 3.94 for the UKP dataset and 4.67 for the IBM one.

Example 4.2 *Original sentence: It is possible to fuel nuclear power plants with other fuel types than uranium.*
*Adversarial attack: It is **totally** possible to fuel nuclear power plants with other fuel types than uranium.*

Nouns Similar to the adjectives method we proposed, this list-based attack exchanges a noun with its hyponym. Again, we only replace one word per perturbation producing one perturbation example for each noun in the sentence. This method generated an average of 12.19 perturbations per sentence on the UKP dataset, whilst the number increases to 17 for the IBM dataset.

Example 4.3 *Original sentence: When it comes to infertile couples, should not they be granted the **opportunity** to produce clones of themselves?*
*Adversarial attack: When it comes to infertile couples, should not they be granted the **chance** to produce clones of themselves?*

Conjunctions This method consists of adding adverbial conjunctions, such as *furthermore* or *nonetheless*, at the beginning of the input sentence. If the sentence already begins with an adverbial conjunction, the sentence is skipped. This attack delivers an average of 2.69 perturbations per sentence on the UKP dataset and 2.88 on the IBM.

Speculative Adverbs They are modal adverbs related to the possibility property of verbs. This method is similar to the aforementioned scalar adverbs perturbation. Another list-based attack where modal adverbs related to the possibility property of verbs, such as *certainly*, are added directly before a verb. In this case, we obtained an average of 1.67 perturbations per sentence on the UKP dataset and 1.75 on the IBM.

Example 4.4 *Original sentence: Even the gateway effect — the theory that cannabis leads to other drugs — was discarded long ago.*
*Adversarial attack: Even the gateway effect — the theory that cannabis **indeed** leads to other drugs — was discarded long ago.*

Topic Alternatives Previous work has shown that including the topic in the BERT input increases the performance of the model [13]. Thus, exchanging the topic with alternatives is a relevant perturbation to evaluate. For each topic in the two corpora, we created a list of alternatives. For example, *arms limitation* for *gun control* or *capital punishment* for *death penalty*. While we created an average of 4.25 alternatives per topic for UKP dataset, for the IBM dataset on average, there were 2.75 alternatives per topic.

4.3. User Study: Quality of Generated Perturbations

As an additional evaluation criteria of the generated perturbations, we conducted a user study about the preservation of semantics between the original sentence and the sentence after the modification. Both versions of a sentence were presented to the user and the user was asked if the two sentences 1) have the same meaning, 2) do not share the same meaning, or 3) if the transformed sentence is not meaningful, where "not meaningful" could mean either that the sentence has become ungrammatical or that it does not make sense anymore. For each answer option there was also a text field giving the possibility to voluntarily provide a justification of their decision. In total, 72 pairs of sentences were presented to each participant comprising every type of perturbation, but the topic alternative and punctuation deletion. We excluded the topic alternatives from the study, because the topic is an independent part of the model input and does not modify the grammaticality or semantics of a sentence. Same holds for the deletion of punctuation, which only changes the semantics of a sentence in some rare case of rhetorical questions. Moreover, the participant thinking of proper punctuation might have shifted their focus from the actual task, i.e. semantic similarity. The sentence length of each pair of sentences was controlled to have a difference of maximum one standard deviation from the mean sentence length of the sentences in the dataset. Participants in the user study were mainly non-native speakers with a higher educational degree (Masters degrees or Ph.D.) and a fluent level of English. In total, 31 people completed the questionnaire.

The perturbation method with the highest percentage of preserving the meaning of the sentence, i.e. 93.68%, is adding conjunctive adverbs. Naturally, this barely impacts the meaning of a single sentence. For the NE replacement, 71.3% of the people found the exchange as meaningful. The main criticism was that the new named entity, especially when they were acronyms, was unknown to the participant. Overall, employing word embeddings as a distance criteria to select NEs of the same type preserves the meaningfulness in most cases. Replacing an adjective with its synonym was in 61.04% of the cases found to be meaningful. While for the other cases, it was reported that the selected synonym was not suitable for the given context. Similar feedback was gathered for the hyponym replacement of nouns. Here, in 52.53% of the cases the selected noun did not fit the context, as either being too specific or unrelated to the topic. Inserting speculative adverbs was perceived as not changing the meaning of a sentence in 57.82% of the cases. A main observation reported by the participants is the change in credibility or certainty of the mentioned studies and other evidence, e.g. changing facts to opinions. Indeed, this does change the semantics of a sentence, but with respect to an argument classifier the uncertainty of an evidence does not matter as much as that it is correctly detected as being an argument. From this point of view, despite the study results, we consider this perturbation method a valid and meaningful transformation. Compared with the other perturbation types, adding and replacing scalar adverbs caused with 57.33% the most cases of changes of a meaning of a sentence. The participants found that this transformation often breaks the grammaticality of a sentence. A future challenge is to find the right place to insert such adverbs, because some of them can either precede the target word or come only after it. Moreover, one has to consider if a target word can scale. For example, *genetic, mandatory* or *guilty* cannot be compared. There is no such thing as *fairly mandatory*. These points need to be address in future work.

5. Results and Discussion

In this section, we present and discuss the results of our two lines of experiments. First, the success rates for each perturbation type, and second, the adversarial training.

5.1. Adversarial Attacks

Table 1 reports on the success rate (the percentage) of adversarial examples over the total of generated perturbations.

Perturbation Type	UKP all			UKP x-topic			IBM x-topic	
	Arg+	Arg-	NoArg	Arg+	Arg-	NoArg	Arg	NoArg
Named Entities	7.06	7.30	2.02	6.14	7.22	2.30	1.51	0.18
Adjectives	10.90	10.02	6.70	12.16	10.37	5.89	3.79	0.03
Punctuation	8.86	9.74	4.21	10.41	10.61	4.34	2.78	0.19
Scalar Adverbs	5.87	7.15	3.41	7.39	7.57	3.29	2.01	0.08
Nouns	13.91	14.56	7.35	15.08	14.65	7.6	8.43	0.53
Spec. Adverbs	6.31	6.89	2.99	7.49	6.82	2.53	1.42	0.06
Conjunctions	5.87	7.29	4.33	9.66	9.52	4.56	3.64	0.4
Topic Alternatives	0.81	1.33	0.29	1.07	1.13	0.41	1.14	0.08

Table 1. Label-wise success rate of each perturbation type on the different test scenarios.

Looking at the in-domain test scenario, i.e., UKP all, one can observe that the Arg- label is more affected by the attacks than the Arg+ label, with exception of the adjectives. The adjective and noun replacement have the highest success rates in attacking the models. For adjectives, this could be explained with the fact that they usually carry sentiments whose perception might differ if they appear in a pro or con argument. For nouns, the replacement with hyponyms has the highest success rate, but given that in the human evaluation only in 47.47% of the cases the perturbation was perceived as meaningful, we cannot consider results with respect to this perturbation as fully reliable.

Overall, the positive classes, Arg+, Arg- and Arg, showed to be more vulnerable to attacks than the no argument class. Usually, the structure of the task at hand, which features in the data one tries to learn, is associated with the positive class. Meaning that the complementary class does not necessarily contain a distinctive pattern in the feature space, because it contains everything which is not wanted. Hence, it cannot be as efficiently attacked as the learnt patterns for the positive classes. Unexpectedly, deleting the punctuation resulted in a comparatively high success rate. After reviewing the attention scores of the model, we found that, contrary to our expectations, the model tends to attend to punctuation. This observation needs to be confirmed at a larger scale, though. Exchanging the topic with alternative wording resulted in an insignificant success rate not affecting the model. Concerning the cross topic evaluation, the UKP x-topic shows partially higher vulnerability than its in-domain counterpart. Since cross domain is the harder task, the confidence scores are lower for unseen test data, and with that the overall performance compared to in-domain models. A less confident model is easier to attack, explaining the higher success rates. Interestingly, the IBM x-topic is not as vulnerable to attacks as the UKP x-topic model. Again, as can be noticed in Table 2, the overall performance of the IBM model is higher. Since in both cases the same model architecture is employed, the only difference is the data. The IBM dataset seems to be more structurally uniform than the UKP dataset, explaining why test performance is higher and the success rate of attacks lower. Another point supporting this is that the exchange of NEs, which the IBM corpus contains more per sentence than the UKP one, barely changes the classification of an input example. This connotes that, in the case of the IBM data, NEs are not as important for the model justifying that they can be exchanged without losing the argumentative function of a sentence. Even though this further justifies our named entity perturbation method, it is ineffective in this case. Overall, BERT-based topic-dependent argument classification models are relatively robust against minor changes to the input, but still vulnerable to a certain degree. In roughly 5-10% of the cases, adding a meaning preserving word changes the prediction of the model.

5.2. Adversarial Training

The most common strategy to defend from adversarial attacks and make a model more robust is adversarial training. This is covered in our second line of experiments, whose results are reported in Table 2.

	UKP all	UKP x-topic	IBM x-topic
standard training	73.70	60.9	77.58
adversarial training	80.22	59.3	78.57

Table 2. Results in macro f_1 for models with and without adversarial training.

For the in-domain scenario (UKP all), one can observe an increase of 6.5 points in f_1-score compared to the model trained without adversarial examples. This shows that adding linguistic variants of the training data helps in predicting unseen test data from the same domain. Intuitively this makes sense, arguments are often rephrased differently or are re-used as targets for undercutting, for example. With respect to BERT, this raises questions. In the aforementioned experiments on perturbation efficiency, we have seen that BERT seems to be quite robust against our adversarial attacks. Also, in previous works, models based on language model pre-training advanced the state-of-the-art, which was said to be due to the natural language understanding capabilities learnt during pre-training. Accordingly, this should mean that slight variations of the input are covered by the language model. The increase in performance with adversarial training shows that this supposed NLU capability is either not fully utilized or blurred during fine-tuning, or was limited in the first place. We assume it is a mixture of both, since other experiments in different domains show that BERT-like models are more robust than recurrent networks [5], but also that the language modelling capabilities of self-attentive models are limited [12,19]. Even if the success rates of our perturbations are only between 5-10%, added up these make quite a number of examples, which BERT is vulnerable to. Adding these linguistic variations to the training data, though, boosts the NLU capabilities making the model more receptive for them. Note that this way the training data is increased by roughly a factor of twenty. This indeed shows that adversarial training helps in-domain predictions and improves the robustness of a model, as intended. Table 3 shows examples where adversarial training corrected the model prediction.

A justified doubt coming up here is the question of overfitting. *Did the adversarial training really help in NLU or did it just improve learning the dataset?* In the latter case,

topic	sentence	$pred_1$	$pred_2$
gun control	Five women are murdered with guns every day in the United States.	NoArg	Arg+
school uniforms	Up to now , this uniform is still in use , making it the ‘ oldest uniform in history. ’	Arg+	NoArg
cloning	I find this reasoning absolutely ridiculous, since a person is a person despite their genetic source or if artificially created.	Arg-	Arg+

Table 3. Examples were adversarial training improved the model prediction. $pred_1$ model prediction before adversarial training, $pred_2$ model prediction after adversarial training, which is also the true label.

one would see a decline in cross domain evaluation, because the model is overly focused on in-domain specific features. As can be seen in Table 2, the cross domain performance is not dropping significantly with adversarial training. Both models are still in an acceptably similar range compared with their normally trained counterpart. The UKP x-topic losses 1.6 f_1-score, while the IBM model even shows a slight increases of roughly 1 f_1-score. Meaning that the generalizability of the models is preserved, ergo they did not overfit on the training domain. So *why is it that adversarial training helps in-domain, but does not improve the cross domain performance?* At this point, we like to repeat the aforementioned distinction between robustness and generalizability. For us, robustness is more related to the ability to understand language in the sense of linguistic flexibility; being able to understand differently worded phrases about the same thing. Generalizability, on the other hand, is the ability of a model to transfer and apply already learnt patterns to a new domain. In our case, an increase in performance for the models tested on cross topics is related to the generalizability. While depending on the task of the application field, generalizability and robustness have a strong overlap, we think, one has to carefully distinguish them for argument mining. Usually, cross domain in AM means that the model should be able to detect arguments for a topic unseen during training. Assuming the new topic is not somehow related to the topics seen during training, this means, the model has to infer everything associated with a given input sentence and decide if this can be an argument related to the topic or not. The problem is one can only conditionally infer new arguments from existing arguments in the semantic space. If the two arguments are structurally similar to a certain degree (or use similar key components), it is possible. But finding new arguments for an unseen domain is beyond language modelling. It requires also a deep understanding of knowledge and common sense. Especially the latter two cannot be efficiently learnt from word co-occurrences alone [19,10]. As a result, it is not surprising that augmenting training data with alternative wording of the data does not improve generalizability. After all, the examples added for adversarial training are mostly noise with respect to the new unseen test domain; noise, which is not negatively affecting the generalizability of the BERT model.

6. Conclusion

This paper presents the first approach to test the robustness of argument classification models through adversarial examples. We investigate different ways to produce meaningful adversarial examples, and we assess their quality through a user study. Furthermore, we demonstrate the effectiveness of adversarial training and we empirically show that it helps to improve robustness without impacting generalizability. Obtained results highlight that BERT is robust but still vulnerable to simple changes to the input.

For the future, a further evaluation of the robustness of argument classification models is needed. This goes beyond the weaknesses of the here presented approach, such as controlling the selection of synonyms and hyponyms or the positioning and selection algorithm for adverbs. Combinations of different perturbation types are worth exploring. As well as white-box approaches [5], where the target words are carefully selected dependent on model parameters. Another highly interesting and relevant field is the evaluation of paraphrases as a means to attack models. As a more general goal, experiments are required to find the right balance between augmenting the training data with adversarial examples and noise for efficient adversarial training.

References

[1] M. Alzantot, Y. Sharma, A. Elgohary, B.-J. Ho, M. Srivastava, and K.-W. Chang. Generating natural language adversarial examples. In *Proc. of EMNLP 2018*, pages 2890–2896, 2018.

[2] E. Cabrio and S. Villata. Five years of argument mining: a data-driven analysis. In *Proc. of IJCAI 2018*, pages 5427–5433, 2018.

[3] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proc. of NAACL-HLT 2019*, pages 4171–4186, 2019.

[4] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *Proc. of ICLR 2015*, 2015.

[5] Y.-L. Hsieh, M. Cheng, D.-C. Juan, W. Wei, W.-L. Hsu, and C.-J. Hsieh. On the robustness of self-attentive models. In *Proc. of ACL 2019*, pages 1520–1529, 2019.

[6] M. Iyyer, J. Wieting, K. Gimpel, and L. Zettlemoyer. Adversarial example generation with syntactically controlled paraphrase networks. In *Proc. of NAACL 2018*, pages 1875–1885, 2018.

[7] R. Jia and P. Liang. Adversarial examples for evaluating reading comprehension systems. In *Proc. of EMNLP 2017*, pages 2021–2031, 2017.

[8] J. Lawrence and C. Reed. Argument mining: A survey. *Computational Linguistics*, 45(4):765–818, 2019.

[9] M. Lippi and P. Torroni. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Techn.*, 16(2):10, 2016.

[10] T. Mayer. Enriching language models with semantics. In *Proc. of ECAI 2020*, 2020.

[11] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proc. of ACM AsiaCCS 2017*, pages 506–519, 2017.

[12] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, and P. J. Liu. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

[13] N. Reimers, B. Schiller, T. Beck, J. Daxenberger, C. Stab, and I. Gurevych. Classification and clustering of arguments with contextualized word embeddings. In *Proc. of ACL 2019*, pages 567–578, 2019.

[14] E. Shnarch, C. Alzate, L. Dankin, M. Gleize, Y. Hou, L. Choshen, R. Aharonov, and N. Slonim. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proc. of ACL 2018*, pages 599–605, 2018.

[15] C. Stab, T. Miller, B. Schiller, P. Rai, and I. Gurevych. Cross-topic argument mining from heterogeneous sources. In *Proc. of EMNLP 2018*, pages 3664–3674, 2018.

[16] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. In *Proc. of ICLR 2014*, 2014.

[17] E. F. Tjong Kim Sang and F. De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. *CoRR*, 2003.

[18] X. Yuan, P. He, Q. Zhu, R. R. Bhat, and X. Li. Adversarial examples: Attacks and defenses for deep learning. *CoRR*, 2017.

[19] R. Zellers, A. Holtzman, Y. Bisk, A. Farhadi, and Y. Choi. HellaSwag: Can a machine really finish your sentence? In *Proc. of ACL 2019*, pages 4791–4800, 2019.

[20] W. E. Zhang, Q. Z. Sheng, A. Alhazmi, and C. Li. Adversarial attacks on deep-learning models in natural language processing: A survey. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 11(3):1–41, 2020.