Models in the Wild: On Corruption Robustness of NLP Systems

Abstract

Natural Language Processing models lack a unified approach to robustness testing. In this paper we introduce WildNLP - a framework for testing model stability in a natural setting where text corruptions such as keyboard errors or misspelling occur. We compare robustness of models from 4 popular NLP tasks: Q&A, NLI, NER and Sentiment Analysis by testing their performance on aspects introduced in the framework. In particular, we focus on a comparison between recent state-of-the-art text representations and non-contextualized word embeddings. In order to improve robustness, we perform adversarial training on selected aspects and check its transferability to the improvement of models with various corruption types. We find that the high performance of models does not ensure sufficient robustness, although modern embedding techniques help to improve it. We release corrupted datasets and code for WildNLP framework for the community.

1 Introduction

Adversarial examples have been shown to severely degrade performance of deep learning models (Goodfellow et al., 2015) (Papernot et al., 2016). Natural Language Processing systems are no different in this respect. Multiple areas of NLP, such as machine translation (Belinkov and Bisk, 2017), question answering (Jia and Liang, 2017), or text classification (Liang et al., 2017) have been studied to assess the impact of adversaries generated with various methods. However, these works tend to focus on one area only, often with attacks designed just for the selected problem. It makes comparisons between models, datasets, and NLP areas impossible. In particular, the robustness of modern word embedding systems - such as ELMo (Peters et al., 2018), Flair (Akbik et al., 2018) and language model based BERT (Devlin et al., 2018) remains unstudied.

In this article, we evaluate the behavior of natural language models in the wild. We propose WildNLP - a systematic and comprehensive robustness testing framework which can be used for any NLP model. Instead of focusing on elaborate attacks, which are unlikely to originate by accident, we measure the quality of models in a natural setting, where input data is poisoned with errors involuntarily generated by actual users. We put these notions into a set of tests called *aspects*. Moreover, we introduce the concept of corruption severity and prove that it is critical to model improvement via adversarial training. The framework is aimed at any NLP problem irrespective of its form of input and output.

In summary, our contributions are the following:

1. **We offer a systematic framework for testing corruption robustness - the WildNLP.** In total, we introduce 11 aspects of robustness testing, with multiple severity levels. We release the code and a collection of popular datasets that are corrupted with WildNLP for the community[1]. The framework is easy to extend. New aspects can be defined by the community.

2. **We test corruption robustness of a number of NLP tasks: question answering (Q&A), natural language inference (NLI), named entity recognition (NER), and sentiment analysis (SA).** We verify stability of models trained on contextualized embeddings like ELMo and Flair in contrast to non-contextualized FastText (Bojanowski et al., 2017) and GloVe (Pennington et al., 2014).

[1] link omitted due to double-blind review process

We also analyze BERT in the task of Q&A. We find that new forms of text representation, despite greater contextual awareness, do not offer a sufficient increase in robustness.

3. **We find that model training on one aspect does improve performance on another aspect, contrary to previous studies** (Belinkov and Bisk, 2017). For this to be true, two corruption types must be similar to some extent.

In section 2 we present related literature in the domain of NLP robustness. In section 3 we present WildNLP framework, describing in detail each introduced aspect. In section 4 we compare robustness of NER, Q&A, NLI and Sentiment Analysis. In section 5 we perform adversarial training on `Qwerty` aspect with different severities and test these models on other aspects. We conclude in section 6.

2 Related Work

The problem of natural noise in textual data has been studied by Belinkov and Bisk (2017), however exclusively in the context of character-based machine translation models. They find that errors such as typos and misspelling cause significant drops in BLEU scores. Other recent approaches to generating textual adversaries include the work of Liang et al. (2017), who exploit important word manipulations for text classification models from 2014 and 2015. Gao et al. (2018) identify important words and apply 4 kinds of character perturbations: swap, substitution, deletion and insertion. They test on vanilla LSTM and CNN model, applying them to 8 datasets. Among others, they aim for the character swaps to map a word vector to an 'unknown' vector in traditional word embeddings. Ribeiro et al. (2018) create rules of substitutions between texts which produce correct and semantically identical samples in Q&A domain. Glockner et al. (2018) design adversaries for NLI systems, swapping words which share a relation such as antonymy or co-hyponymy.

3 WildNLP: Corruption Robustness Testing Approach

We postulate that performance of each model should be tested on three levels:

1. **Performance measures.** Well established metrics such as F1 score, accuracy, BLEU score should indicate to what extent the model performs correctly on the testset.

2. **Corruption robustness.** This is robustness towards corruptions which can occur naturally in the model deployment setting. They reflect involuntary perturbations introduced to text by users, resulting from misspelling, haste or varied writing habits. As such, these are black box attacks as no knowledge of underlying models is exploited.

3. **Targeted robustness.** These are the attacks designed for a specific problem and/or dataset, or demanding access to model internals. An example is the whole class of white box attacks (Ebrahimi et al., 2017) as well as highly specialized attacks (Jia and Liang, 2017).

3.1 Corruption Aspects

The WildNLP aspects define classes of common disturbances found in natural text. These corruptions are produced naturally due to haste, lacking space, individual writing habits or imperfect command of English.

`Articles`. Randomly removes or swaps articles into wrong ones.

`Swap`. Randomly shuffles two characters within a word.

`Qwerty`. Simulates errors made while writing on a QWERTY-type keyboard. Characters are swapped for their neighbors on the keyboard.

`Remove_char`. Randomly removes characters from words.

`Remove_space` Removes a space from text, merging two words.

`Misspelling`. Misspells words appearing in the Wikipedia list of commonly misspelled English words[2].

`Digits2words`. Rewrites digit numbers into words.

`Homophones`. Changes words into their homophones from the Wikipedia list of common misspellings/homophones[3]. The list contains around 500 pairs or triples of homophonic words.

`Negatives`. This aspect reflects attempts made by some Internet users to mask profanity

[2] https://en.wikipedia.org/wiki/Commonly_misspelled_English_words

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

Table 1: Examples of text corruptions introduced by WildNLP aspects.

Aspect	Example sentence
Original	Warsaw was believed to be one of the most beautiful cities in the world.
`Article`	Warsaw was believed to be one of **a** most beautiful cities in world.
`Swap`	Warsaw **aws** believed to be one **fo teh** most beautiful cities in the world.
`Qwerty`	Wa**d**saw was b**d**lieved to be one of the most beautiful citi**e**e in the world..
`Remove_char`	Warsaw was believed to be one **o th** most **eautiful** cities in the world.
`Remove_space`	Warsaw was believed **tobe** one of the most beautiful cities in the world.
Original	You cannot accidentally commit vandalism. Vandalism used to be a rare occurrence.
`Misspelling`	You can not **accidentaly** commit vandalism. Vandalism used to be a rare **occurrance**.
Original	Bus Stops for Route 6, 6.1
`Digits2words`	Bus Stops for Route **six**, **six point one**
Original	Choosing between affect and effect can be scary.
`Homophones`	Choosing between **effect** and effect can **bee** scary.
Original	Laughably foolish or false: an absurd explanation.
`Negatives`	**Laughab*y fo*lish** or **fal*e**: an **a*surd** explanation.
Original	Sometimes it is good to be first, and sometimes it is good to be last.
`Positives`	Sometimes it is **go*d** to be first, and sometimes it is **goo*** to be last.
`Marks`	Sometimes, it is good to be first and sometimes**,** it**,** is good to be last.

or hate speech in online forums to evade moderation. We perform masking of negative words from Opinion Lexicon[4]. The lexicon contains a list of English positive and negative opinion words or sentiment words, in total around 6800 words.

`Positives`. Masks positive words from Opinion Lexicon, similarly as in the case of Negatives (described above).

`Marks`. Randomly removes and insert punctuation marks. Marks are inserted between last letter of a word and space.

The severity of perturbations can be varied. In the case of `Swap`, `Qwerty` and `Remove_char` we control it by defining how many words will be affected. In the case of `Article`, it is defined by a probability of corruption of each article.

Table 1 presents examples of resulting changes for each aspect.

4 Experiments

We test corruption robustness on various NLP tasks and models. Each of the models is run on the specific dataset it has been trained on in the original setting. The datasets are preprocessed by the WildNLP framework to obtain corrupted data with multiple aspects. An important point in the experimental setting is the application of various word embeddings. We focus on testing the robustness of models trained with newly introduced context-aware embeddings: ELMo, Flair and language model based BERT. We compare their performance on corrupted data to older embedding systems - GloVe, FastText (within InferSent) and in the case of one of sentiment analysis models, even one-hot encoded words. We do so to verify the assumption that greater context awareness and lack of problems with out-of-vocabulary (OOV) words in ELMo, Flair and BERT would increase robustness of models.

4.1 Experimental Setting

We use our framework on the selection of well known models that are widely used in NLP community. For training ELMo-based models we use open-source implementations available in AllenNLP (Gardner et al., 2017), for BERT we follow implementation of HuggingFace[5] and for the rest of the models we use original author research code. In particular, following models and datasets are used in experiments:

- **Q&A task**

  Models. We test BiDAF and BERT trained on the SQuAD dataset (Rajpurkar et al., 2016). We analyze two versions of BiDAF - with ELMo (BiDAF-E) and GloVe (BiDAF-G) embeddings. BiDAF uses character and word embeddings with a bidirectional attention flow to obtain a query-aware context representation. BiDAF is one of the state-of-the-art models on the SQuAD leaderboard. On the other hand, BERT applies a bidirectional Transformer to language modeling task and is currently used with great success in various NLP tasks, achieving the new state-of-the-art.

[4] https://www.cs.uic.edu/~liub/FBS/sentiment-analysis.html

[5] https://github.com/huggingface/pytorch-pretrained-BERT

Question&Answer	Context
Q: How many provinces did the Ottoman empire contain in the 17th century? **A:** 32	(...) At the beginning of the 17th century the empire contained 32 provinces and numerous vassal states.(...)

Table 2: Exemplary context, question and answer from SQuAD dataset.

Premise	Hypothesis	Type
A woman with a green headscarf, blue shirt and a very big grin.	The woman is young.	neutral
An old man with a package poses in front of an advertisement.	A man poses in front of an ad.	entailment
A couple walk hand in hand down a street.	A couple is sitting on a bench.	contradiction

Table 3: Exemplary hypotheses, questions and answers from SNLI dataset.

Token	SOCCER	-	JAPAN	GET	LUCKY	WIN	CHINA	IN	DEFEAT	.
Class	O	O	I-LOC	O	O	O	I-PER	O	O	O

Table 4: Exemplary tagged sentence fron CoNLL dataset.

Review	Sentiment
Kutcher played the character of Jake Fischer very well, and Kevin Costner played Ben Randall with such professionalism. The sign of a good movie is that it can toy with our emotions. (...)	positive
Once again Mr. Costner has dragged out a movie for far longer than necessary. Aside from the terrific sea rescue sequences, of which there are very few I just did not care about any of the characters. (...)	negative

Table 5: Excerpts from exemplary reviews from IMDB dataset.

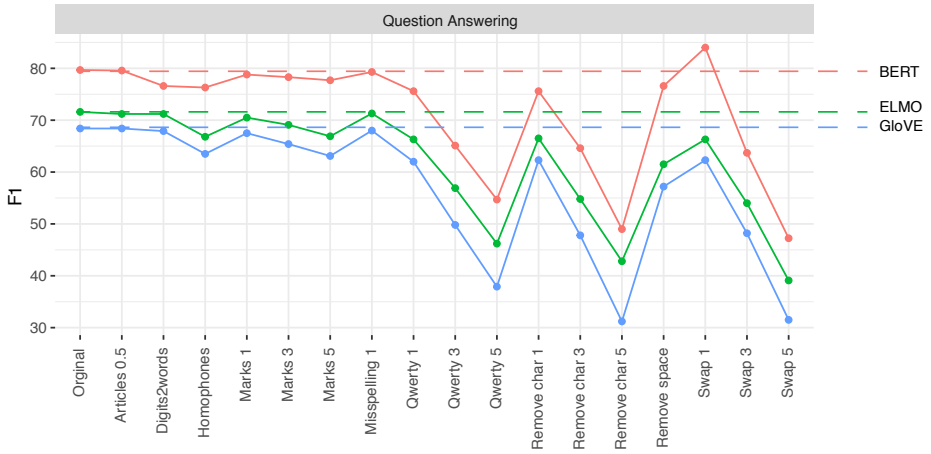

Figure 1: Robustness testing results for Q&A models.

Table 6: Influence of the severity of corruption of training data on results of corrupted testsets in Q&A BiDAF ELMo model.

	Trainset		
	Original	Qwerty_1	Qwerty_5
Tested on	EM	EM	EM
Original	**71.6**	70.7	69.0
Qwerty_1	66.4	**68.2**	67.8
Qwerty_5	46.2	58.2	**63.8**

We evaluate the models with the common performance scores in Q&A task, which are Exact Match (EM) and F1 score.

Dataset. SQuAD dataset comprises around 100,000 question-answer pairs prepared by crowdworkers. The dataset is based on Wikipedia articles. Table 2 displays examples of the question-answer pairs.

- **NLI task**

Models. We analyze decomposable attention model (Parikh et al., 2016) trained on ELMo embeddings and InferSent model (Conneau

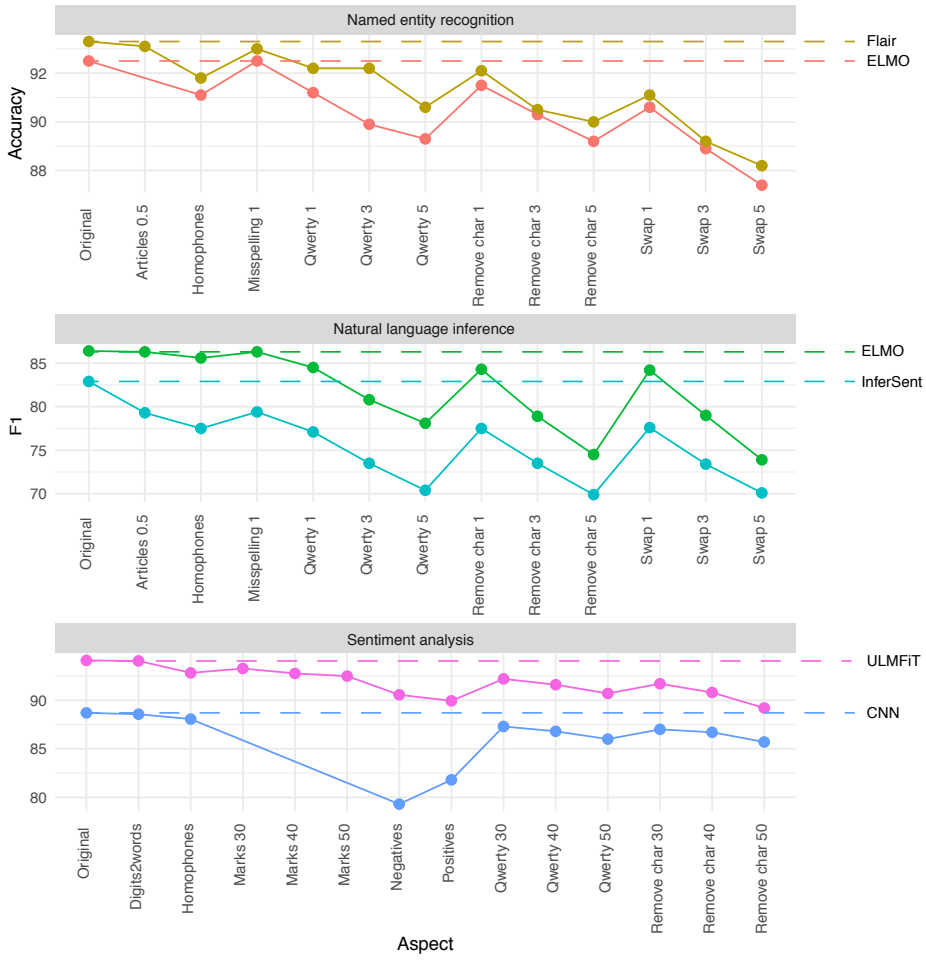

Figure 2: Robustness testing results for NLI, NER, and SA models. Corruptions of smaller severities (1-5) are not evaluated on SA models due to greater length of IMDB dataset sequences.

et al., 2017). The aim of InferSent embeddings is to create the universal sentence representations. They are initialized with FastText embeddings and trained using SNLI dataset. It was an effort to create pretrained and universal sentence representations that could be transfered to a variety of tasks.

Dataset. The Stanford Natural Language Inference (SNLI) Corpus (Bowman et al., 2015) is a collection of 570,000 manually created and labeled English sentence pairs. Table 3 contains an example of the three possible entailment relations.

- **NER task**

  Models. We use two sequence tagging models with ELMo implementation (CRF-E) (Peters et al., 2018) and Flair (Akbik et al., 2018). Flair comprises new word embeddings an a BiLSTM-CRF sequence labeling system. It models words as sequences of characters, which allows to effectively eliminate the notion of separate tokens. Flair is currently the state-of-the-art model in NER task.

  Dataset. The CoNLL 2003 dataset is a standard training dataset used in NER sequence tagging. It is a collection of news articles from Reuters corpus annotated as Person, Organization, Location, Miscellaneous, or Other for non-named entities. Due to licensing agreement this is the only corrupted dataset that we cannot release.

- **SA task**

  Models. We use the current state-of-the-art ULMFiT model (Howard and Ruder, 2018) that consists of language model pretrained on Wikipedia and fine-tuned on the specific text corpus that is used in classification task. In adversarial training scenario, we pretrain this language model on corrupted data. We

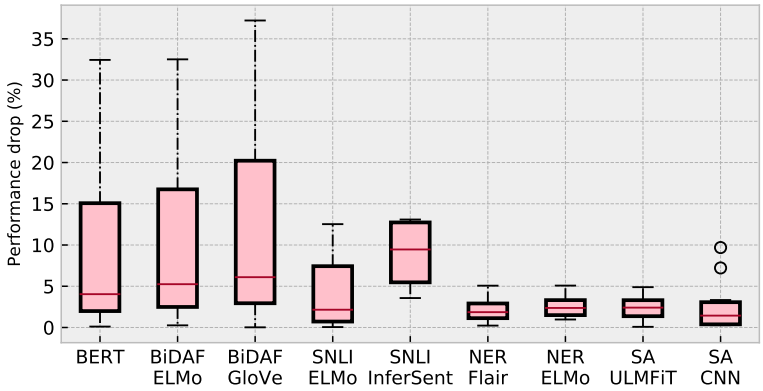

Figure 3: Variances of performance drops of models for all perturbations shown in Table 1 and Table 2. For Q&A models, EM measure is displayed.

compare ULMFiT with CNN based classification model, which uses one-hot encoding of words.

Dataset. We train and test described models on IMDB dataset that consists of 25000 positive and 25000 negative reviews of movies.

4.2 Model Robustness

Figure 1 (Q&A models) and Figure 2 (other models) present aggregate results of testing on all models and all corruption aspects. Variability and scale of performance drops are depicted in Figure 3. Tables with full results can be found in Appendix A.

Robustness measure. To comprehensively measure model robustness to corruptions, we calculate an overall mean of drops across all aspects (Av-Drop). We use this aggregated metric to compare robustness between models.

Q&A. The robustness of Q&A models was the lowest of all tested tasks. The corruptions which proved most damaging to the performance and in result to Av-Drop were the following: `Swap_5` (32 - 37 EM drop), `Remove_char_5` (29 - 37 EM drop), `Qwerty_5` (25 - 30 EM drop).

BERT and ELMo-based systems were found to mitigate performance loss to some degree compared to GloVe. However, their performance loss pattern across corruptions was similar to GloVe, and the difference of Av-Drop between BERT (most robust model) and BiDAF GloVe (least robust model) was 2.8 pp, despite huge performance differences reflected in F1 and EM (1).

We observe that severity of aspects plays an important role in drop of performance metrics across all Q&A models. For aspects that corrupt individual words like `Qwerty`, `Remove_char` or `Swap`, drop in performance of GloVe-based models is intuitive - we substitute words from out of vocabulary (OOV) with *unknown token*. However, in the case of ELMo and BERT the problem of OOV tokens is not that severe - they are character or subword-based. Still, we observe an average drop of F1 metric on these three aspects (severity 5) at the level of 23.04 (BiDAF-E) and 24.46 (BERT) in comparison to drop of BiDAF-G at 32.9. Lower severities of word corruptions induce much lower drops - in case of severity 1 it is still a noticeable difference of 4.48 (BiDAF-E), 3.44 (BERT) and 5.63 (BiDAF-G).

WildNLP also tests on aspects that do not alter words but sentences. As previously, we state that context-aware models should be indifferent to such changes as they do not alter sentence meaning. However, we observe that aspects such as `Remove_space` and `Marks` decrease F1 values among all Q&A even by 8.89 in case of `Remove_space` tested with BiDAF-E, whereas BERT proves to be more robust to this sentence-level corruption with drop of F1 at 2.47.

NLI. Natural Language Inference task tested by WildNLP framework is more robust when trained with decomposable attention model with ELMo embeddings (Dec-E) rather than simple MLP classifier that uses sentence embeddings created by InferSent method (InferSent). The Av-Drop for Dec-E is half the value of Av-Drop for InferSent, being at the level of 4.19. On all sets of aspects, Dec-E model has lower drops of performance metric. However, it still has relatively high drops when it comes to word corruption aspects like `Qwerty`,

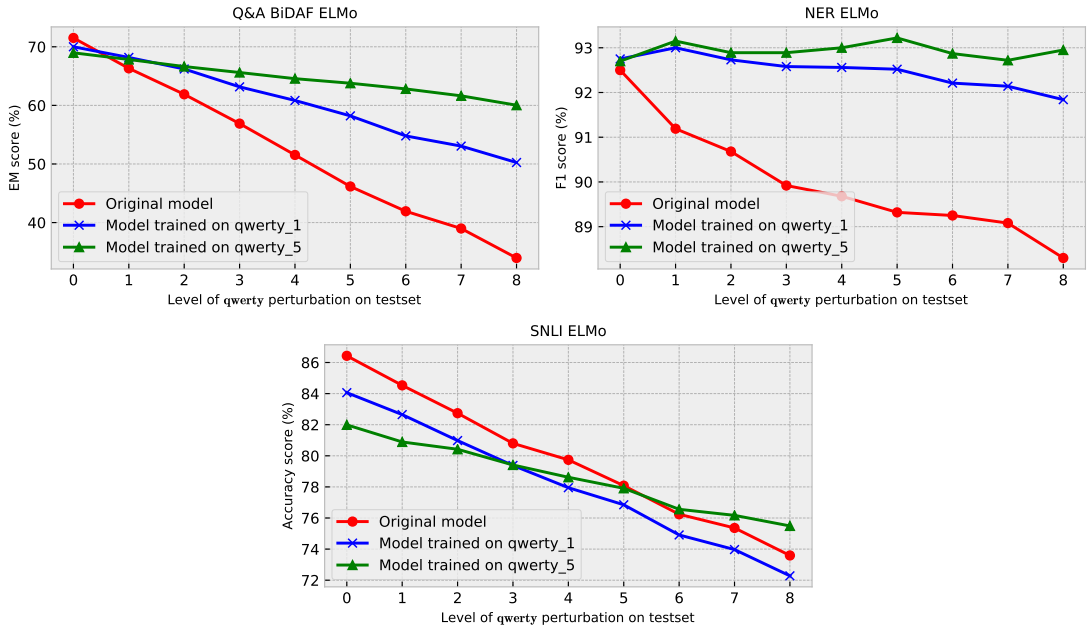

Figure 4: Performance of models trained on varied levels of `Qwerty` aspect tested on varied levels of `Qwerty` aspects applied to testset.

`Remove_char` or `Swap`, with average drop of 10.92 at severity 5 and 2.09 at severity 1. InferSent performs worse by around 3 pp (5.56 and 12.82 respectively).

However, when we consider sentence level aspects like adding extra commas to the sentence, Dec-E model is very robust, having only 0.85 of drop in accuracy on highest possible severity.

NER. Both NER models seems to be robust, having the Av-Drop measure at the level of 2.37 (CRF-E) and 2.14 (Flair). However, in the case of state-of-the-art NER models, differences in performance are so small, that such relatively small values of Av-Drop can be seen as high. As we processed only non-NE words with WildNLP framework, we assume that model predictions of named entities are dependent on surrounding context words.

SA. ULMFiT model was found to be slightly less robust than CNN using one-hot encodings (2.36 vs 2.28 of Av-Drop). Drop in performance of the CNN model was mainly caused by `Positives` and `Negatives` corruptions (7.22 and 9.7 Av-Drop) which can be observed as the two outliers in Figure 3. Presumably this behavior is caused by the model's focus on detecting sentiment-carrying words, which were on average rarely affected by other corruptions. On the other hand, ULMFiT was less affected by `Positives` and `Negatives` corruptions (3.6 and 4.2 Av-Drop) probably because of its reliance on context and more subtle expressions of sentiment. In spite of the fact that the CNN model suffered from out-of-vocabulary words problem (corrupted words were simply unrecognized) while ULMFiT did not, the CNN proved more robust to most deformations in WildNLP framework.

5 Robustness Enhancements

We use adversarial training to research the potential of overcoming corruption errors. We validate two hypotheses:

1. **Adversarial training on data corrupted with aspects of greater severity helps to resolve problems with data corrupted with lesser severity.** For example, training on `Qwerty_5`-corrupted data should increase performance of data corrupted with `Qwerty_1` up to `Qwerty_5` severities.

2. **Adversarial training on one corruption type should increase model robustness to other corruptions.** However, Belinkov and Bisk (2017) suggest that this is not the case. They find that models trained on one type of noise do not perform well on others in character-based translation models. Based on this, we hope to prove that robustness can be

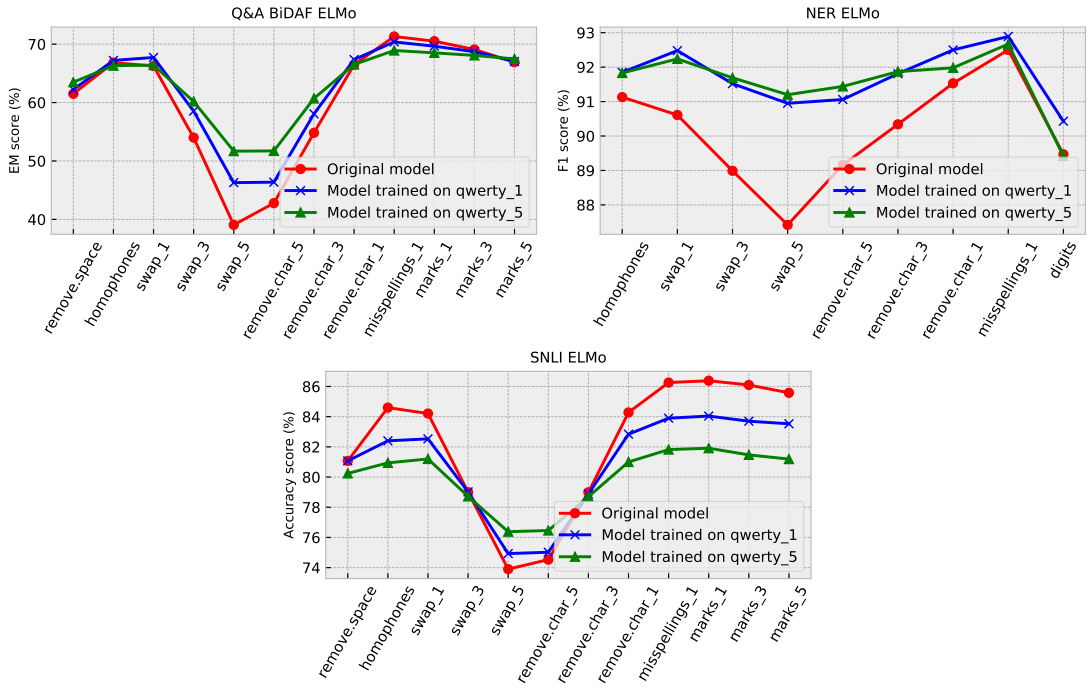

Figure 5: Influence of training on data corrupted with `Qwerty` aspects on testing on other aspects.

improved between aspects which are related.

Corruption severity. In agreement with our hypothesis we find that increased severity of corruption during training does increase performance on data corrupted with the same aspect type but lesser severity. Table 6 presents numeric scores for the training setting in Q&A BiDAF ELMo models, while Figure 4 shows plots for multiple models. In all scenarios, we test on `Qwerty_1` and `Qwerty_5` corruptions.

Interestingly, in the case of NER models, results obtained on models trained on both corruption types are even better than for the original model (for `Qwerty_5` model, this behavior is consistent across levels of severity of test data perturbations).

Empirically, the severity of `Qwerty` perturbation (and others) does make the text unintelligible for humans at some point. For example, this boundary was found to be level 5 for Q&A questions. However, the Q&A BiDAF ELMo model trained on `Qwerty_5` performs reasonably well even at severity level 8. This suggests that the model learned to decode this corruption even beyond human ability.

Relation between corruption types. To verify relations between performance of models trained and tested on various corruption types, we test models trained on `Qwerty` corruption with severity 1 and 5. `Qwerty` exhibits similarities to `Swap` and `Remove_char` types, since all of them imply manipulations of word characters. We observe that Bidaf ELMo and NER ELMo models trained on `Qwerty` and tested on similar aspects perform better than original models not trained in adversarial setting. Results are depicted in Figure 5.

6 Conclusions

In this work, we have presented the WildNLP framework for corruption robustness testing. We have introduced 11 text corruption types (at various severity levels) which can occur naturally in model deployment setting: misspellings, keyboard errors, attempts at masking emotional language, and others. We test on four NLP areas and 12 models in total, verifying corruption robustness of state-of-the-art BERT system and new LM-based embeddings: ELMo and Flair, contrasted with GloVe and Fasttext. We find that the problem of lacking corruption robustness is not solved by these recent systems. However, we find that the issue can be partially alleviated by adversarial training, even across aspects. We believe that problem of adversarial examples in NLP is still vague and hard to quantify. Without doubt, more work is needed to make models robust to natural noise, whether by robust word embeddings, model architectures, or better datasets.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649.

Yonatan Belinkov and Yonatan Bisk. 2017. Synthetic and natural noise both break neural machine translation. *CoRR*, abs/1711.02173.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching word vectors with subword information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for NLP. *CoRR*, abs/1712.06751.

Ji Gao, Jack Lanchantin, Mary Lou Soffa, and Yanjun Qi. 2018. Black-box generation of adversarial text sequences to evade deep learning classifiers. *CoRR*, abs/1801.04354.

Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. 2017. Allennlp: A deep semantic natural language processing platform.

Max Glockner, Vered Shwartz, and Yoav Goldberg. 2018. Breaking NLI systems with sentences that require simple lexical inferences. *CoRR*, abs/1805.02266.

Ian Goodfellow, Jonathon Shlens, and Christian Szegedy. 2015. Explaining and harnessing adversarial examples. In *International Conference on Learning Representations*.

Jeremy Howard and Sebastian Ruder. 2018. Fine-tuned language models for text classification. *CoRR*, abs/1801.06146.

Robin Jia and Percy Liang. 2017. Adversarial examples for evaluating reading comprehension systems. *CoRR*, abs/1707.07328.

Bin Liang, Hongcheng Li, Miaoqiang Su, Pan Bian, Xirong Li, and Wenchang Shi. 2017. Deep text classification can be fooled. *CoRR*, abs/1704.08006.

Nicolas Papernot, Patrick D. McDaniel, and Ian J. Goodfellow. 2016. Transferability in machine learning: from phenomena to black-box attacks using adversarial samples. *CoRR*, abs/1605.07277.

Ankur P. Parikh, Oscar Täckström, Dipanjan Das, and Jakob Uszkoreit. 2016. A decomposable attention model for natural language inference. In *EMNLP*.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392. Association for Computational Linguistics.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 856–865. Association for Computational Linguistics.