Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020), pages 1882–1894
Marseille, 11–16 May 2020

Stress Test Evaluation of Transformer-based Models in Natural Language Understanding Tasks

Carlos Aspillaga*[†], Andrés Carvallo[†][‡], Vladimir Araujo*[†][‡]
[†]Pontificia Universidad Católica de Chile, Santiago, Chile
[‡]IMFD, Santiago, Chile
{cjaspill, afcarvallo, vgaraujo}@uc.cl

Abstract

There has been significant progress in recent years in the field of Natural Language Processing thanks to the introduction of the Transformer architecture. Current state-of-the-art models, via a large number of parameters and pre-training on massive text corpus, have shown impressive results on several downstream tasks. Many researchers have studied previous (non-Transformer) models to understand their actual behavior under different scenarios, showing that these models are taking advantage of clues or failures of datasets and that slight perturbations on the input data can severely reduce their performance. In contrast, recent models have not been systematically tested with adversarial-examples in order to show their robustness under severe stress conditions. For that reason, this work evaluates three Transformer-based models (RoBERTa, XLNet, and BERT) in Natural Language Inference (NLI) and Question Answering (QA) tasks to know if they are more robust or if they have the same flaws as their predecessors. As a result, our experiments reveal that RoBERTa, XLNet and BERT are more robust than recurrent neural network models to stress tests for both NLI and QA tasks. Nevertheless, they are still very fragile and demonstrate various unexpected behaviors, thus revealing that there is still room for future improvement in this field.

Keywords: adversarial evaluation, stress tests, natural language inference, natural language understanding, question answering

1. Introduction

Deep learning has allowed for solving several problems related to natural language processing (NLP), even outperforming human performance in some tasks, such as multi-label classification (Tsoumakas and Katakis, 2007), document screening (Carvallo and Parra, 2019), named entity recognition (Nadeau and Sekine, 2007), among others. However, previous research has shown that neural networks are powerful enough to memorize the training data, which limits their ability to generalize or to really understand the tasks they are dealing with (Zhang et al., 2017).

One way to test NLP models is by using adversarial tests, which implies an intentional perturbation of the input sentence to confuse a model into making wrong predictions. This methodology has shown that models are still weak (Belinkov and Bisk, 2018; Iyyer et al., 2018; Ribeiro et al., 2018; Ebrahimi et al., 2018). Other researchers have also shown that language models can "falsely" solve the task. In other words, they might be taking advantage of dataset failures or artifacts on the input sentences in order to guess the answer (Gururangan et al., 2018; Agrawal et al., 2016; Levy et al., 2015). These evaluations, also known as "stress tests", have been performed on classic models based on recurrent networks (RNN). However, Transformer-based models such as RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) and BERT (Devlin et al., 2018), which are state-of-the-art for NLU tasks, have not been systematically evaluated under severe stress conditions. Only BERT has been tested with similar objectives as ours (Hsieh et al., 2019; Jin et al., 2019; Niven and Kao, 2019), but not in a systematic way as here nor in the same scenarios.

In this work, we focus on three language models based on the state-of-the-art Transformer architecture (RoBERTa, XLNet and BERT), with the aim of carrying out a stress test evaluation on two natural language understanding (NLU) tasks. On the one hand, Natural Language Inference (NLI), also known as recognizing textual entailment (RTE) which consists of finding semantic relations between a premise sentence and an associated hypothesis, by **classifying** if they are entailed, in contradiction or in neutral relationship. On the other hand, we apply stress tests on a question-answering (QA) task, also known as machine reading comprehension (MRC) which consists of **predicting** the answer to a question given a paragraph.

The evaluation of the NLI task was performed using the MultiNLI dataset (Williams et al., 2018) following the methodology of Naik et al. (2018). For the QA task we used the SQuAD dataset (Rajpurkar et al., 2016) and adversarial techniques introduced by Jia and Liang (2017). We also developed a new adversarial dataset for SQuAD, using techniques inspired on Belinkov and Bisk (2018)[1].

All our test procedures try to prove the strength of the models, by distracting, confusing or proving their competence. Experiments show that all models are affected by stress tests, but on Transformer-based models, the adversaries have smaller impact compared to previous models based on RNNs. This behavior could be explained by the large number of parameters and their prior training. Nevertheless, in this work we not only measure the impact on performance of various adversarial or noisy conditions, but also reveal that in some cases the state-of-the-art models behave in strange and unexpected ways.

We provide detailed quantitative analysis on all the performed tests, and in some cases we report representative examples via inspection of the attention matrices that these models produce during inference when tested under adversarial test scenarios.

* Equal contribution, listing order is random.

[1] we released the dataset at `https://github.com/caspillaga/noisy-squad`

2. Transformer for Natural Language Understanding

The Transformer (Vaswani et al., 2017) is a deep learning architecture originally proposed for neural machine translation applications. The main idea behind this model is the multi-head self-attention, the ability to attend to different parts and aspects of the input sequence to compute a contextual representation of it, at increasing levels of abstraction (layers). This architecture allows surpassing long-term dependency problems that are common on Recurrent Neural Networks (RNN) models, and adding the possibility of being highly parallelizable.

Early works such as GPT (Radford and Sutskever, 2018) and BERT (Devlin et al., 2018) proposed variants of the Transformer architecture for language modeling (Bengio et al., 2001). These works show that the representations learned on large-scale language modeling datasets are effective for downstream sentence-level tasks (i.e. NLI) and token-level tasks (i.e. QA) via fine-tuning. However, compared to RNNs, no systematic evaluation of robustness and failure modes for these kind of models (specially the most recent variants) have been performed in previous works.

In this work, we evaluate three state-of-the-art models on their large version: BERT (Devlin et al., 2018), which was the first model to introduce bidirectional representation in the Transformer encoder and masked modeling, XLNet (Yang et al., 2019) that proposed the permutation modeling to prevent the corruption of the input with masks, and RoBERTa (Liu et al., 2019), which can be seen as a BERT optimization that includes additional pre-training and hyperparameter improvements.

We use the HuggingFace python library (Wolf et al., 2019), which includes pre-trained models, in order to fine-tune each model to a classifier for the NLI task and a regressor for the QA task. We used the hyperparameters specified in the original paper for each model, to achieve an accuracy close to the ones reported for each task.

Additionally, we include pre-Transformer baselines as a comparison reference. These models rely on the LSTM architecture (Hochreiter and Schmidhuber, 1997) and are task-dependent. However, our analysis and discussion are mainly about experiments on Transformer-based models.

3. NLI Task Description

3.1. Task

The MultiNLI corpus (Williams et al., 2018) is a crowdsourced collection of 433k sentence pairs annotated with textual entailment information from a broad range of genres. In this task, given a *premise*, the model has to determine whether a *hypothesis* is true (entailment), false (contradiction), or undetermined (neutral).

3.2. Baselines

As a baseline to evaluate stress test performance for this task, we chose the winner of RepEval 2017 Shared Task (Nangia et al., 2017), which proposed a model of stacked BiLSTMs with residual connections (Nie and Bansal, 2017). Also, we used the baseline proposed in the original paper (Williams et al., 2018) of the dataset, which consists of a standard BiLSTM.

4. QA Task Description

4.1. Task

SQuAD, the Stanford Question Answering Dataset (Rajpurkar et al., 2016) is a widely used Question Answering benchmark that consists of a collection of English Wikipedia paragraphs with more than 100k associated question-answer pairs generated via crowdsourcing. The task is designed in a way that the solution to each question is literally contained in the corresponding paragraph, so the task is to predict the answer text span in the corresponding passage. We use SQuAD v1.1 instead of SQuAD v2.0 to allow comparability with previous work.

4.2. Baselines

To be consistent with previous work, we used BiDAF (Seo et al., 2016) and Match-LSTM (Wang and Jiang, 2016) as baselines to compare stress tests against Transformer-based models. BiDAF consists of embedding, attention and modeling layers with a BiLSTM, that outputs a vector with information of the context and the query, and finally an output layer with probabilities indicating where the answer starts and ends in the context text. In the case of Match-LSTM, the model is an architecture that remembers important word-level matching results to get better predictions of the answers.

5. Experiments

5.1. NLI Task Evaluation

Our experiments on the MultiNLI dataset closely follow the Naik et al. (2018) procedure, which conducted a stress test evaluation of several models of the RepEval 2017 Shared Task. Below we describe each test set[2] used in this work and Table 1 shows some examples, however for further details of the sets construction we refer the readers to the work by Naik et al. (2018).

5.1.1. Distraction Test

The distraction test explores the model robustness after a text with a clear "True" value is added.

- One way to evaluate this is by decreasing the lexical similarity between *premise* and *hypothesis*. On the one hand, the *word overlap* set adds a tautology ("and true is true") at the end of each *hypothesis* sentence. On the other hand, the *length mismatch* set adds five times the same tautology to each *premise*.

- We can also evaluate this by the inclusion of strong negations. The *negation* set is quite similar to the previous ones, but in this case, the tautology added to the *hypothesis* includes negation words ("and false is not true").

5.1.2. Noise Test

This test verifies the model strength against noisy data, in terms of *spelling errors*. It has two types of permutations on

[2] We use the sets provided by the authors to avoid discrepancy during the procedure. `abhilasharavichander.github.io/NLI_StressTest`

Test Set	Premise	Hypothesis
Word Overlap	Then he ran.	He ran like an athlete and true is true.
Length Mismatch	Then he ran and true is true and true is true and true is true and true is true and true is true.	He ran like an athlete.
Negation	Then he ran.	He ran like an athlete and false is not true.
Spelling Errors	Then he ran.	He ran like an athleet.
Antonymy	The Joint Venture had justified itself by failure.	The Joint Venture had justified itself by success.
Numerical Reasoning	Adam spent 1/6 of his lifetime in adolescence.	Adam spent less than 1/6 of his lifetime in adolescence.

Table 1: Examples of stress tests for the NLI task.

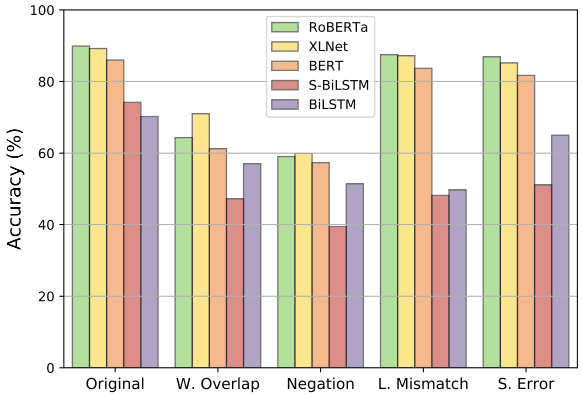

Figure 1: Accuracy results in the development set and adversarial sets: word overlap, negation, length mismatch and spelling error. Only matched partition is shown.

a word randomly selected from the *hypothesis*: swap of adjacent characters within the word, and random substitution of a character next to it on the English keyboard. Note that only one substitution is performed for the entire sentence.

5.1.3. Competence Test

The competence test consists of two evaluation sets to measure the reasoning ability of the models.

- Understanding of *antonymy relationships*. This set includes sentences that result in contradiction simply by using an antonym in some adjectives or nouns.

- *Numerical reasoning* ability of a model. This evaluation includes statements of simple algebraic problems with solutions as *premises*. The entailed, contradictory and neutral hypotheses were generated through the use of heuristic rules.

5.2. NLI Task Results

Table 2 shows the results of the performed tests. It can be seen that all models decrease their accuracy in all evaluations. However, Transformer-based models show more robustness in some tests. The analysis of the results of the models in each stress test is shown on the following sections.

5.2.1. Models Performance on Distraction Test

Figure 1 shows a bar graph of the "matched" partition of the evaluation sets on the different types of distraction tests. As mentioned in a previous section, the distraction tests allow us to check the robustness in two different ways.

On the one hand, the effect of introducing *negation* words drops the models performance below 60% of accuracy, close to the baselines. We checked the model predictions on the negation test v/s the development set and we found that BERT and XLNet obtained 93% and 91% of E-N (entailment predicted as neutral) error respectively. In contrast, RoBERTa obtained 85% of N-E error (neutral predicted as entailment). This could occur due to the introduction of extra negation words ("false" and "not").

On the other hand, the decrease of lexical similarity by *word overlap* and *length mismatch* evaluation shows:

- In the first case (*word overlap* set), the Transformer-based models reach around 60% accuracy, which is approximately 20% less than in the development set. We found a similar behavior with the previous set (*negation*), where BERT and XLNet obtained 83% and 61% of E-N error respectively. It also stands out that RoBERTa achieved 89% of N-E error.

- In the second case (*length mismatch*), the models performed better than expected, because they reached almost the same accuracy as in the development set. We hypothesize that these results may be due to the *length mismatch* set modifying the *premise* sentence instead of the *hypothesis* as in the negation of the *word overlap* sets, which suggests that in order to answer, the model is paying more attention to that sentence.

To verify the results on the *length mismatch* set, we extended the evaluation by testing the addition of the tautology "and true is true" in the *hypothesis* or in the *premises* N times (where $N = 1..5$). Figure 2 shows the performance of XLNet in these tests, likewise we observed similar behavior on the other models. We noticed that the inclusion of the distractions to the premise sentence does not affect the model performance. However, when we add the tautology a single time (which is equivalent to the *word overlap* test) to the *hypothesis* sentence, the performance drops about 20%, and the more repetitions we add, the more accuracy increases, almost reaching the same performance ob-

Model	Original Dev		Distraction Test						Noise Test		Competence Test		
			Word Overlap		Negation		Length Mismatch		Spelling Error		Antonymy		Numerical Reasoning
	M	MM	M	MM	M	MM	M	MM	M	MM	M	MM	
RoBERTa	**90.0**	**89.7**	64.3 [28.5]	62.3 [30.5]	59.0 [34.4]	58.5 [34.8]	**87.5** [2.8]	**88.2** [**1.7**]	**85.3** [**5.2**]	**85.7** [**4.5**]	63.9	59.2	64.9
XLNet	89.2	89.1	**71.0** [20.4]	**68.9** [22.7]	**60.0** [32.7]	**59.5** [33.2]	87.2 [**1.9**]	87.5 [1.8]	83.5 [6.4]	83.7 [6.1]	74.7	70.9	63.9
BERT	86.0	86.1	61.2 [28.8]	56.8 [34.0]	57.3 [33.4]	57.6 [33.1]	83.7 [2.7]	84.6 [**1.7**]	79.5 [7.6]	79.8 [7.3]	64.6	59.2	56.8
S-BiLSTM	74.2	74.8	47.2 [36.4]	47.1 [37.0]	39.5 [46.8]	40.0 [46.5]	48.2 [35.0]	47.3 [36.8]	51.1 [31.1]	49.8 [33.4]	15.1	19.3	21.2
BiLSTM	70.2	70.8	57.0 [**18.8**]	58.5 [**17.4**]	51.4 [**26.8**]	51.9 [**26.7**]	49.7 [29.2]	51.2 [27.7]	65.0 [7.4]	65.1 [8.1]	13.2	9.8	31.3

Table 2: Classification accuracy (%) of Transformer-based models and baselines. Both genre-matched (M) and mismatched (MM) sets were evaluated. Values in brackets represent the percentage of reduction with respect to the original dev set.

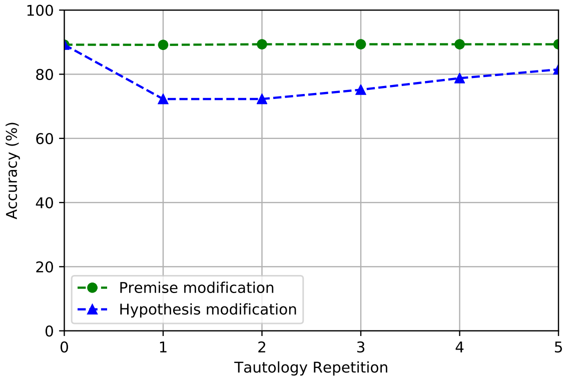

Figure 2: Accuracy (%) of XLNet after the addition of different number of tautologies in hypothesis or premises.

tained in the development set. We also checked the attention weights, and did not identify anomalous behavior.

The unexpected result in accuracy indicates that the lexical similarity is not a strong enough signal to generate distraction in this type of model, as they can discern the tautologies anyway. Moreover, the model seems to pay more attention to the *hypothesis* sentence in order to respond, without discarding the *premise*. However, the distraction evaluation indicates that these Transformer-based models are fragile to adversarial attacks that include strong negation words.

5.2.2. Models Performance on Noise Test

The noise test with the *spelling error* set exhibits that Transformer-based models perform very well. They only lose between 2 to 5 percentage points in accuracy with respect to the *development* set. The results suggest that the multi-head self-attention mechanism of these models is very effective at recovering the global information from the corrupted sentence.

However, the adversarial attacks of this set only modify one word of the *hypothesis*. This explains why there is no sudden drop in performance in models, even for the BiLSTM-based models.

5.2.3. Models Performance on Competence Test

As we supposed, Transformer-based models work quite well in this evaluation task. In the case of the *antonymy* test, the models exceeded baselines by approximately 50 percentage points in accuracy. This is probably because Transformers were pre-trained on a diverse and big corpus, allowing them to adequately represent the majority of the words of the dictionary. XLNet and BERT were trained with BookCorpus and Wikipedia, so we expected better accuracy of RoBERTa which used additional data. However, XLNet outperformed others by at least 10 percentage points, suggesting that permutation modeling could help capture antonymy relationships better.

Furthermore, the results on the *numerical reasoning* evaluation show a lower performance for all models. In this task, XLNet and RoBERTa have similar accuracy but have different behavior. On the one hand, XLNet specialized in classifying “entailment” samples, achieving 90% in that class. On the other hand, RoBERTa specialized in “neutral” category, obtaining 89% of correct answers. In both cases, the remaining classes achieved less than 74% of accuracy (the model finds it hard to distinguish between those classes). These results indicate that Transformer-based models trained in the NLI task have serious difficulties in numerical reasoning and that they take different strategies to solve the task.

For both evaluations, we also explored the attention weights via the *BertViz* library (Vig, 2019). Appendix B shows a brief analysis of some specific cases on all the mentioned Transformer-based models.

5.2.4. Annotation Artifacts Exploitation Test

Gururangan et al. (2018) found that MultiNLI dataset has annotation artifacts. It means that crowd workers who participated in the creation of the data, adopted heuristics to generate the *hypothesis* in an easy and fast way. For instance, they usually use some keywords such as “not”, “never”, etc. to create negation sentences.

To evaluate if Transformer-based models leverage the artifacts, we tested the models by removing the *premise* sentence in the development set. In other words, the models

	Matched	Mismatched
Majority Class	35.4	35.2
RoBERTa	35.2	35.8
XLNet	35.4	35.8
BERT	35.5	35.7
S-BiLSTM	45.2	45.4
BiLSTM	37.4	38.3

Table 3: Performance (%) of premise-unaware text models on MultiNLI development set. Greater accuracy means more exploitation of artifacts, thus smaller numbers mean the models performed best.

are unaware of the *premises* of the dataset.

Table 3 shows the results of this experiment. It is possible to see that Transformer-based models perform similar to the majority class[3], which denotes an unbiased guess of the models. In contrast, BiLSTM-based models show significant proportion of correctly classified samples without even looking at the *premise* (which is an undesirable behavior). This result demonstrates that Transformer-based models are in fact learning to take into account and relate the two sentences of the NLI task in order to choose the correct answer, which is consistent with the findings in Section 5.2.1.

5.3. QA Task Evaluation

One of our test scenarios was taken from Jia and Liang (2017), which intentionally adds a new adversarial sentence at the end of SQuAD passages of the development set. These sentences are especially designed (via different strategies) to act as a decoy to confuse the model. The other test scenario is inspired on Belinkov and Bisk (2018). Although originally proposed for a different task, we replicated the 5 types of noise proposed by the authors, and applied them on the development set of SQuAD.

5.3.1. Adversarial Sentence Tests

In Jia and Liang (2017), the authors proposed 4 strategies to create a sentence especially designed to confuse models by pretending to be the correct answer to a specific question, although they are unrelated with the question. This adversarial sentence is concatenated to the corresponding paragraph provided at test time. The 4 strategies proposed were:

- **AddOneSent**: Adjectives and nouns of the question are replaced by antonyms. Named entities and numbers are replaced by their nearest word in GloVe (Pennington et al., 2014). This modified question is then turned into declarative form (using a set of manually defined rules) and a fake answer of the same type as the original answer is inserted. Finally the sentence is manually checked and fixed via crowdsourcing.

- **AddSent**: Identical to *AddOneSent* but generating multiple candidate sentences (adversaries) and keeping only the one that induces the biggest error when tested on a specific model.

Article: Super Bowl 50
Context: Peyton Manning became the first quarterback ever to lead two different teams to multiple Super Bowls. He is also the oldest quarterback ever to play in a Super Bowl at age 39. The past record was held by John Elway, who led the Broncos to victory in Super Bowl XXXIII at age 38 and is currently Denver's Executive Vice President of Football Operations and General Manager. *Quarterback Jeff Dean had jersey number 37 in Champ Bowl XXXIV.*
Question: What is the name of the quarterback who was 38 in Super Bowl XXXIII?
Original prediction: John Elway
Prediction after adversarial phrase is added: Jeff Dean

Figure 3: An example of an *AddOneSent* adversarial sample. This example was taken from Jia and Liang (2017). In this case we can see that the model correctly answered the original question, but after the inclusion of the adversarial sentence (in *italic blue*), the model fails (answer in red).

- **AddAny**: The adversarial sentence is generated by sampling random words and successively replacing them by elements from a sampled set of 20 words each time. Words are selected from this set by using a criterion that tries to minimize the confidence of the model on the correct answer. The 20-word set is sampled from a list of common words plus the words from the question. This process is repeated iteratively 6 times for each adversarial phrase.

- **AddCommon**: Identical to *AddAny*, but in this case the 20-word set is sampled from the list of common words directly.

5.3.2. Noise Tests

Although originally proposed for a different task, we replicated the 5 types of noise introduced by Belinkov and Bisk (2018). In each experiment, a specific noise type was applied to each word in the passage of SQuAD's development set. The question was kept unchanged, and the answers were adapted to preserve consistency with the modified passage. In contrast to the noise tests performed in the NLI setting (Section 5.1.2), the scenario tested here is significantly more aggressive because it introduces noise to every word in the reference text.

The 5 noise types tested are:

- **Natural Noise**: Words are replaced by real typing errors of people. To automate this, we used a collection of word corrections performed by people in web platforms that keep track of edits history (Max and Wisniewski, 2010; Zesch, 2012; Wisniewski et al., 2013; Šebesta et al., 2017).

- **Swap Noise**: For each word in the text, one random pair of consecutive characters is swapped (e.g. $expression \rightarrow exrpession$).

- **Middle Random Noise**: For each word in the text, all characters are shuffled, except for the first and last characters. (e.g. $expression \rightarrow esroxiespn$).

[3] The majority class is used as a baseline of random guessing.

- **Fully Random Noise**: For each word in the text, all characters are shuffled (e.g. $expression \rightarrow rsnixpoees$).

- **Keyboard Typo Noise**: For each word in the text, one character is replaced by an adjacent character in traditional English keyboards (e.g. $expression \rightarrow exprwssion$).

5.4. QA Task Results

Similarly to the observations for the NLI experiments, for QA it is clear that the performance of all models is affected by the stress tests, with Transformer-based models being the most robust in all the cases analyzed. Detailed results can be found in Table 4.

5.4.1. Results on Adversarial Sentence Tests

Figure 5 shows a bar graph that compares the accuracy of the tested models under the different adversarial strategies. When we analyze the results of the *AddOneSent* experiments, we notice an accuracy reduction between 18.1% and 21.7% for the Transformer-based models, and greater than 39.5% for non-Transformer models. In spite of showing greater robustness in comparison with their counterpart, Transformer-based models still suffer from a significant impact on performance, which elucidates a clear opportunity for future improvements on these kind of models. The same phenomenon is observed for *AddSent* adversaries, but more pronounced (as expected, since *AddSent* tests the worst case for each candidate question). We see accuracy reductions ranging from 27.7% and 32.2% for Transformer-based models, and greater than 54.6% for non-Transformer models.

We notice that as the model is more powerful in the main task (accuracy in the unmodified SQuAD v1.1 development set), it also achieves greater robustness. This conclusion is hopeful because other works have asserted that more powerful models could justify their performance on their higher memorization capabilities (Zhang et al., 2017). These experiments, in contrast, indicate that the models are improving their reading capabilities in a balanced fashion.

Interestingly, *AddAny* and *AddCommon* adversaries show that those strategies are very model-specific, as evidenced by the fact that Transformer-based models only reduce their accuracy in small degree when tested against adversaries where other architectures failed. These results are relevant because, as reported by Jia and Liang (2017), those adversaries (and especially *AddAny*) turned to be very effective when trying to mislead the models that they were targeting. This cross-check between different model's adversaries for *AddAny* is consistent with the results reported by Jia and Liang (2017), although in the case of Transformer-based models, the before-mentioned behavior is even more pronounced. For the case of *AddCommon*, in the other hand, this tests were not reported in previous work nor analyzed by the authors that proposed these adversaries, thus this finding is especially relevant.

Further details on the results of every experiment performed can be found in Appendix A. Also in Appendix C we perform a more qualitative analysis of the attention matrices that these models produce during inference.

Article: Genghis Khan
Context: (...) Maluqi, a tsuretd lteneitnau, was given cmmnoad of the Monogl focres angisat the Jin dytasny whlie Gneghis Kahn was ftgniihg in Ctneral Aais, and Stbuaui and Jbee were aeolwld to prusue the Great Raid itno the Cuaucsas and Kaiven Rus', an idea tehy had peestrned to the Kaaghn on tehir own ieivtnitia. Whlie grnniatg his gneaelrs a gerat dael of amotonuy in mkiang canommd diesscion, Gnhgeis Kahn also epecxted uvwannrieg layolty from them.
Question: Who was delegated command of the Mongol forces against the Jin dynasty?
Answer: Maluqi

Figure 4: A QA adversarial example after the introduction of *Middle Random* noise. Note that only the context (and the answer, accordingly) is modified, but not the question.

5.4.2. Results on Noise Tests

As shown in Figure 6, all five types of noise have a significant negative impact on accuracy on all the tested models. The accuracy reduction is more prominent than on Adversarial Sentence tests (Section 5.4.1) due to the aggressiveness of the strategies tested here.

Swap Noise has a significant impact on accuracy, between 46.3% and 59.0% (for the Transformer-based models) and of 70.1% for Match-LSTM, although only a single pair of characters per word are altered. Performance is only slightly better than when using *Middle Random Noise* (and in that scenario, all the characters are shuffled, except for the first and last ones). We hypothesize that this is due to the fact that by introducing this change, the resulting tokenization differ significantly from the original ones and are also very different from the ones seen in training or fine-tuning, and thus the model is not prepared to answer accurately.

Note also that, in absolute terms, under *Middle Random* noise, the model is still able to correctly answer one in four questions, even though the text is severely transformed (example in Figure 4).

Another unexpected pattern that these tests showed is the fact that for Transformer-based models, the *Keyboard Typo* noise is more challenging to deal with than *Swap Noise*. This finding is especially intriguing because *Keyboard Typo* noise corrupts only one character for each word, and *Swap Noise* corrupts two. For this reason, this result is opposed to what we expected and reveals that *swapping* operations affect these models less than *replacement* operations. This effect may be caused by the fact that the tokenized representation of words with *swapped* characters might be closer to the original one (in the embedding space of each model), or maybe it is because this kind of noise might be more frequent in real misspellings than keyboard typos, so the models were more exposed to this kind of noise during pretraining. Further study is required to find out which phenomenon is the dominant one in this case, but this analysis is out of the scope of this work.

Similarly to what was reported in Belinkov and Bisk (2018), *Natural Noise* is significantly more straightforward to overcome than the other four tested noise types, even considering that in the dataset we built for *Natural Noise*, we forcefully replaced every word by a noisy version of it

Model	Original Dev	Concatenative Adversaries				Noise Adversaries				
		AddOne-Sent	AddSent	AddAny	Add-Common	Swap	Middle Random	Fully Random	Keyboa-rd Typo	Natural
RoBERTa	**85.8**	**70.3** [**18.1**]	61.5 [28.3]	77.3 [9.9]	**84.3** [**1.7**]	**46.1** [**46.3**]	**32.2** [**62.5**]	3.3 [96.2]	**30.4** [**64.6**]	54.9 [36.0]
XLNet	85.2	67.7 [20.5]	**61.6** [**27.7**]	**78.8** [**7.5**]	83.0 [2.6]	43.0 [49.5]	31.9 [62.6]	4.4 [94.8]	27.2 [68.1]	**57.4** [**32.6**]
BERT	82.5	64.6 [21.7]	55.9 [32.2]	71.4 [13.5]	81.1 [**1.7**]	33.8 [59.0]	28.6 [65.3]	**5.5** [**93.3**]	23.1 [72.0]	47.7 [42.2]
Match-LSTM	60.8	30.0 [50.7]	24.8 [59.2]	35.7 [41.3]	52.5 [13.7]	17.8 [70.7]	20.2 [66.8]	4.1 [**93.3**]	9.4 [84.5]	19.7 [67.6]

Table 4: Exact match (%) of Transformer-based models and baselines on the SQuAD v1.1 dev set. AddAny and AddCommon report the worst accuracy after running against all the alternative adversarial datasets of that specific type. For fair comparison, experiments on the adversaries generated for the model itself are excluded in those two specific cases. Values in brackets represent the percentage of reduction with respect to the original dev set.

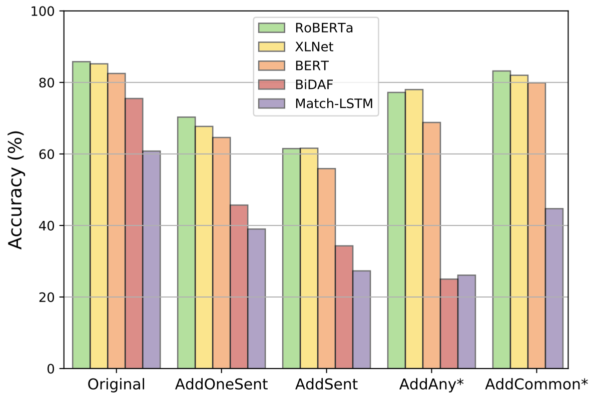

Figure 5: Accuracy results in the adversarial sets proposed by Jia and Liang (2017). AddAny* and AddCommon* report the worst accuracy after running against all the alternative adversarial datasets of that specific type. For fair comparison, experiments on the adversaries generated for the model itself are excluded in those two specific cases.

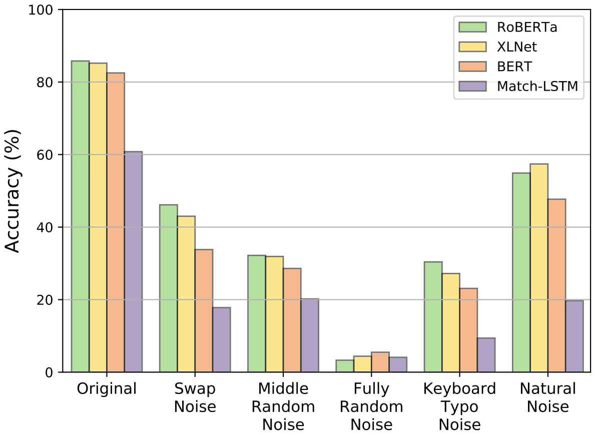

Figure 6: Accuracy results in SQuAD when the models are exposed to noise tests. It is clear that all noise types heavily affect the performance of all the models. Further comparative analysis (Section 5.4.2) show some interesting and unexpected findings in these results.

(when real typing errors were available). It is natural to think that in real scenarios, misspelled words will appear at a much lower rate than in this test. Thus this result can be seen as a kind of lower-bound estimator for performance on *Natural Noise* in real scenarios. When we compare the result of the *Natural Noise* experiments with those of the *Swap Noise* experiments, we hypothesize that the gap in favor to *Natural Noise* is because, during the pre-training phase, the model observed this type of noise (in real occurrences) and was, therefore, able to learn useful representations both for well-written words and for versions with common misspellings.

6. Related Work and Discussion

Prior work (Smith, 2012) discusses the importance of evaluation frameworks that allow characterizing model successes and failures. During previous years, several approaches to test NLP models have been proposed on various tasks, showing that most of the time, predictions are memorized without really understanding the real meaning of utterances (Zhang et al., 2017).

Early research demonstrated that NLP models are fragile to input perturbations. Some attempts at performing stress tests on machine translation systems demonstrated that by adding small perturbations on the input text, the general performance of language models could be profoundly affected (Belinkov and Bisk, 2018; Ribeiro et al., 2018; Ebrahimi et al., 2018). In the same line, the inspiring work of Jia and Liang (2017) proposed an evaluation procedure for language models using the SQuAD dataset. They used SQuAD samples, concatenating adversarial sentences at the end of the paragraph that contains the answer, and showed that 14 open-source models failed when these changes are introduced.

Other relevant findings reveal that models take advantage of lexical cues of the dataset, allowing them to solve the problem falsely. Gururangan et al. (2018) observed that some NLI datasets have annotation artifacts that models exploit to predict the answer without even seeing the rest of the sentence. The same problem was found in the Visual Question Answering (VQA) field. Agrawal et al. (2016) analyzed the behavior of three models based on CNN, LSTM, and attention mechanism by adding adversaries only to the caption of the image, obtaining that most of the times models were paying attention to the text and not the image at inference time.

The success of language models based on the Transformer architecture in tasks such as machine translation (Vaswani et al., 2017; Vaswani et al., 2018), text summarization (Kroening et al., 2008), reading comprehension (Dehghani et al., 2018), among others, motivated new research. Recent works have performed adversarial testing of BERT in Argument Reasoning Comprehension Task (Niven and Kao, 2019). They have shown that tested against adversaries, BERT outperforms BiLSTM and Bag of Vectors baselines, but still has trouble with logic understanding. Furthermore, Jin et al. (2019) showed that BERT is the language model that best performs under adversary attacks when compared to CNN and LSTM in terms of success rate and perturbation rate, preservation of semantic content, and efficiency for text classification tasks. Hsieh et al. (2019) also studied BERT and compared it with recurrent architectures, inspecting the attention matrices of the models and proposing an algorithm to generate adversaries focusing on distracting models but not humans.

Although there is considerable progress in this area, it can be seen that this article differentiates from previous works by systematically evaluating adversaries, artifacts and various severe stress conditions on the state-of-the-art language models based on Transformer (BERT and the models that came after it), in order to verify their language comprehension capabilities and generalization power.

As a final thought, to use these models in real-world applications, the reader must take the conclusions exposed in this work carefully, as some of the noise types and adversaries are far more aggressive than what can be expected in real scenarios. Rather than defining a realistic test scenario, the purpose of this work was to study these models robustness under severe stress conditions to elucidate their strengths and weaknesses, and in some cases quantify an upper bound in the impact that noise or misleading information can have in them. Additionally, some of the adversarial datasets of this work could be used to improve the robustness of the models through an adversarial training process, as they can be seen as an exaggerated version of common typographical errors made by humans.

7. Conclusion

We conducted a stress test evaluation for Transformer-based language models in NLI and QA tasks. In general, our experiments indicate that applying stress tests influenced the performance of all models, but as expected, more recent models such as XLNet and RoBERTa are more robust, showing a better response to this evaluation.

In the NLI task, we verified that the distraction test significantly reduces the performance of all models, especially in the negation test. However, tests on noise examples show that models are somewhat robust, possibly because they were pre-trained in a huge corpus that may have had natural noise. Due to the same reason, the models show good performance for antonymy relationship. Besides, the annotation artifacts test showed that these models take both sentences into account to perform the entailment task and do not take advantage of the artifacts contained in the dataset.

Moreover, in the QA task, experiments revealed that all models suffer in performance when tested with adversarial or noisy samples. Despite this, Transformer-based models turned out to be more robust than their predecessors. We compared Transformer-based models against each and observed that while improving in the main task, models also improved in their robustness in a balanced way. We also noticed that some adversaries are model-specific, as they affect one model but not the rest. Specifically, in the noise tests, we observed that the robustness trend also holds, but noticed some unexpected behavior in relative analysis, as some types of noise affect the models more severely than others, thus revealing specific weak points across all Transformer-based models that did not seem evident at first sight.

We consider this evaluation to be valuable to the community because it exhibits some strengths and weaknesses of the state-of-the-art models. We argue that it is vital that models pass behavioral checks to ensure proper performance in extreme scenarios, where data failures are not being considered. Taking this into consideration, we see that there is still room for future improvements on Transformer-based models.

8. Acknowledgements

We would like to thank Alvaro Soto and Denis Parra for helpful comments. We are also grateful to the anonymous reviewers for their valuable feedback on an earlier version of this paper. This work has been partially funded by Millennium Institute for Foundational Research on Data (IMFD).

9. Bibliographical References

Agrawal, A., Batra, D., and Parikh, D. (2016). Analyzing the behavior of visual question answering models. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1955–1960, Austin, Texas, November. Association for Computational Linguistics.

Belinkov, Y. and Bisk, Y. (2018). Synthetic and natural noise both break neural machine translation. In *International Conference on Learning Representations*.

Bengio, Y., Ducharme, R., and Vincent, P. (2001). A neural probabilistic language model. In T. K. Leen, et al., editors, *Advances in Neural Information Processing Systems 13*, pages 932–938. MIT Press.

Carvallo, A. and Parra, D. (2019). Comparing word embeddings for document screening based on active learning.

Dehghani, M., Gouws, S., Vinyals, O., Uszkoreit, J., and Kaiser, Ł. (2018). Universal transformers. *arXiv preprint arXiv:1807.03819*.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Ebrahimi, J., Lowd, D., and Dou, D. (2018). On adversarial examples for character-level neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 653–663, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Gururangan, S., Swayamdipta, S., Levy, O., Schwartz, R., Bowman, S., and Smith, N. A. (2018). Annotation artifacts in natural language inference data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 107–112, New Orleans, Louisiana, June. Association for Computational Linguistics.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Hsieh, Y.-L., Cheng, M., Juan, D.-C., Wei, W., Hsu, W.-L., and Hsieh, C.-J. (2019). On the robustness of self-attentive models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1520–1529, Florence, Italy, July. Association for Computational Linguistics.

Iyyer, M., Wieting, J., Gimpel, K., and Zettlemoyer, L. (2018). Adversarial example generation with syntactically controlled paraphrase networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1875–1885, New Orleans, Louisiana, June. Association for Computational Linguistics.

Jia, R. and Liang, P. (2017). Adversarial examples for evaluating reading comprehension systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2021–2031, Copenhagen, Denmark, September. Association for Computational Linguistics.

Jin, D., Jin, Z., Zhou, J. T., and Szolovits, P. (2019). Is bert really robust? a strong baseline for natural language attack on text classification and entailment.

Kroening, D., Sharygina, N., Tonetta, S., Tsitovich, A., and Wintersteiger, C. M. (2008). Loop summarization using abstract transformers. In *International Symposium on Automated Technology for Verification and Analysis*, pages 111–125. Springer.

Levy, O., Remus, S., Biemann, C., and Dagan, I. (2015). Do supervised distributional methods really learn lexical inference relations? In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 970–976, Denver, Colorado, May–June. Association for Computational Linguistics.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Max, A. and Wisniewski, G. (2010). Mining naturally-occurring corrections and paraphrases from Wikipedia's revision history. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Languages Resources Association (ELRA).

Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Naik, A., Ravichander, A., Sadeh, N., Rose, C., and Neubig, G. (2018). Stress test evaluation for natural language inference. In *The 27th International Conference on Computational Linguistics (COLING)*, Santa Fe, New Mexico, USA, August.

Nangia, N., Williams, A., Lazaridou, A., and Bowman, S. (2017). The RepEval 2017 shared task: Multi-genre natural language inference with sentence representations. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 1–10, Copenhagen, Denmark, September. Association for Computational Linguistics.

Nie, Y. and Bansal, M. (2017). Shortcut-stacked sentence encoders for multi-domain inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 41–45, Copenhagen, Denmark, September. Association for Computational Linguistics.

Niven, T. and Kao, H.-Y. (2019). Probing neural network comprehension of natural language arguments. *arXiv preprint arXiv:1907.07355*.

Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October. Association for Computational Linguistics.

Radford, A. and Sutskever, I. (2018). Improving language understanding by generative pre-training.

Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (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, Austin, Texas, November. Association for Computational Linguistics.

Ribeiro, M. T., Singh, S., and Guestrin, C. (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, Melbourne, Australia, July. Association for Computational Linguistics.

Šebesta, K., Bedřichová, Z., Šormová, K., Štindlová, B., Hrdlička, M., Hrdličková, T., Hana, J., Petkevič, V., Jelínek, T., Škodová, S., Janeš, P., Lundáková, K., Skoumalová, H., Sládek, Š., Pierscieniak, P., Toufarová, D., Straka, M., Rosen, A., Náplava, J., and Poláčková, M. (2017). CzeSL grammatical error correction dataset (CzeSL-GEC). LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

Seo, M., Kembhavi, A., Farhadi, A., and Hajishirzi, H. (2016). Bidirectional attention flow for machine comprehension. *ArXiv*, abs/1611.01603.

Smith, N. A. (2012). Adversarial evaluation for models of natural language. *arXiv preprint arXiv:1207.0245*.

Tsoumakas, G. and Katakis, I. (2007). Multi-label classification: An overview. *International Journal of Data Warehousing and Mining (IJDWM)*, 3(3):1–13.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Vaswani, A., Bengio, S., Brevdo, E., Chollet, F., Gomez, A. N., Gouws, S., Jones, L., Kaiser, Ł., Kalchbrenner, N., Parmar, N., et al. (2018). Tensor2tensor for neural machine translation. *arXiv preprint arXiv:1803.07416*.

Vig, J. (2019). A multiscale visualization of attention in the transformer model. *arXiv preprint arXiv:1906.05714*.

Wang, S. and Jiang, J. (2016). Machine comprehension using match-lstm and answer pointer. *ArXiv*, abs/1608.07905.

Williams, A., Nangia, N., and Bowman, S. (2018). A broad-coverage challenge corpus for sentence understanding through inference. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1112–1122. Association for Computational Linguistics.

Wisniewski, K., Schöne, K., Nicolas, L., Vettori, C., Boyd, A., Meurers, D., Abel, A., and Hana, J. (2013). MERLIN: An online trilingual learner corpus empirically grounding the European Reference Levels in authentic learner data. URL `https://www.ukp.tu-darmstadt.de/data/spelling-correction/rwse-datasets`.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. (2019). Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). Xlnet: Generalized autoregressive pretraining for language understanding. *arXiv preprint arXiv:1906.08237*.

Zesch, T. (2012). Measuring contextual fitness using error contexts extracted from the Wikipedia revision history. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 529–538, Avignon, France, April. Association for Computational Linguistics.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2017). Understanding deep learning requires rethinking generalization.

Appendix A: Detailed Results on SQuAD Tests

In Table 5, we report the detailed results of the experiments performed on the adversarial versions of the SQuAD dataset using the adversaries proposed by Jia and Liang (2017). In all the experiments, each model was trained/fine-tuned on the original SQuAD v1.1 training set, and tested on each one of the generated adversarial datasets. As a result, we see that all models are affected by these adversarial samples, but also found that some adversaries are model-specific because they do not affect all models as much as they affect the model they are targeting.

	Model under Evaluation				
Targeted Model	**Match-LSTM**	**BiDAF**	**BERT-Large**	**XLNet-Large**	**RoBERTa-Large**
Original (for reference only)	60.8	75.5	82.5	85.2	85.8
AddOneSent	30.0	45.7	64.6	67.7	70.3
AddSent					
Match-LSTM Single	24.8	40.3	62.8	64.6	67.8
Match-LSTM Ensemble	24.2	40.2	62.1	64.4	68.0
BiDAF Single	25.7	34.3	62.3	64.4	67.7
BiDAF Ensemble	25.9	38.3	61.7	64.0	67.6
BERT-Base	26.3	–	60.8	63.3	66.6
BERT-Large	26.9	–	55.9	62.8	66.0
XLNet-Large	27.1	–	61.6	61.6	66.6
RoBERTa-Large	27.6	–	60.9	63.2	61.5
AddAny					
Match-LSTM Single	38.3	57.1	73.8	78.8	78.8
Match-LSTM Ensemble	26.1	50.4	70.7	78.0	77.2
BiDAF Single	43.8	4.8	72.2	79.6	78.4
BiDAF Ensemble	34.7	25.0	68.8	79.1	76.3
AddCommon					
Match-LSTM Single	55.8	–	82.1	83.6	84.6
Match-LSTM Ensemble	44.7	–	81.4	83.0	84.6
BiDAF Single	56.7	41.7	80.9	83.4	84.8
BiDAF Ensemble	52.8	–	79.9	82.0	83.2

Table 5: **Adversarial examples transferability between models**. Each row measures accuracy (%) on adversarial examples designed to attack one particular model. Each column reports the test results of one particular model on all the adversarial datasets.

The information from the first two columns was obtained by running the official implementations used by Jia and Liang (2017). The results are slightly different from the original work because the original weights were not available.

Appendix B: Attention-level Results of NLI Task

Antonymy Evaluation

For this analysis, we took a representative adversarial example where a word in the sentence was replaced by its *antonym*. The model is asked to decide if there is a contradiction, neutral, or entailment relationship between them. We expect the model to connect the attention between the replaced words to predict the correct answer. Assume the following pair of sentences:

```
I saw that daylight was coming, and heard the people sleeping up.
I saw that daylight was coming, and heard the people waking up.
```

In this representative example for testing antonyms, we computed the attentions produced by XLNet, RoBERTa, and BERT. We checked the layers and heads where a clear attention pattern was present between the word and its antonym, as shown in Figures 7 - 9. Within this particular case, for XLNet, we saw that only 2.86% of the total attention heads and layers had this pattern. For RoBERTa, this number was 2.60%, and for BERT 1.56%. On the other hand, for all models, most of the attention was paid to separators and all words from the reference sentence without distinction (Figure 10).

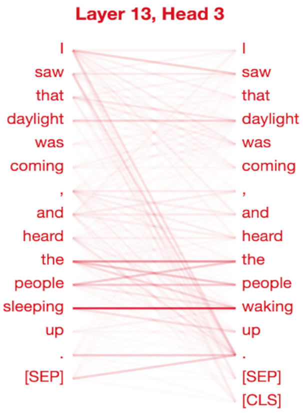

Figure 7: XLNet antonym test

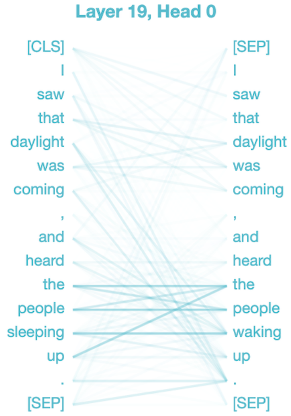

Figure 8: RoBERTa antonym test

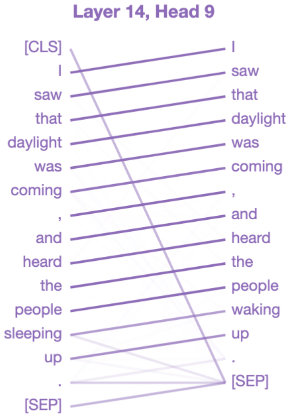

Figure 9: BERT antonym test

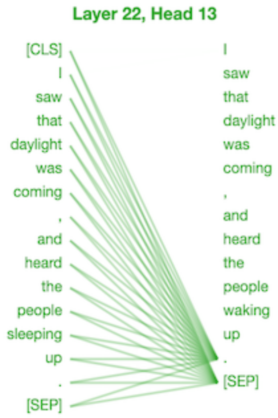

Figure 10: Failed antonym test

Numerical Reasoning Evaluation

For samples of *numerical reasoning* for NLI, the expectation is that the model should pay attention to words like *"more"* or *"less"* to check if there is a change in numerical references. Assume the following pair of sentences:

```
The next day Bob took the test and with this grade, included the new average, was more than 48.
The next day Bob took the test and with this grade, included the new average, was 78.
```

Nevertheless, for this testing example, the *premise* includes *"more than 48"* and the *hypothesis* replaces this last part by *"78"*, but all the models (XLNet, RoBERTa and BERT) incorrectly predicted *"contradiction"*. We observed that the expected pattern (shown in Figures 11- 13) is a very infrequent pattern for all models (for XLNet it appeared in 5.20% of the cases, for RoBERTa in only 4.42% and for BERT this percentage was 1.30%). For other cases, they focused on sentence separators (as shown in Fig 14).

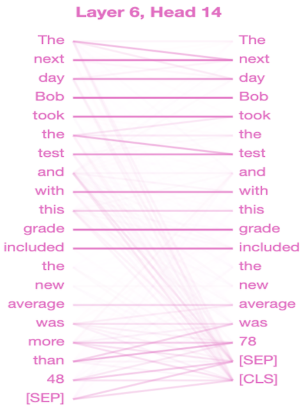

Figure 11: XLNet numerical test

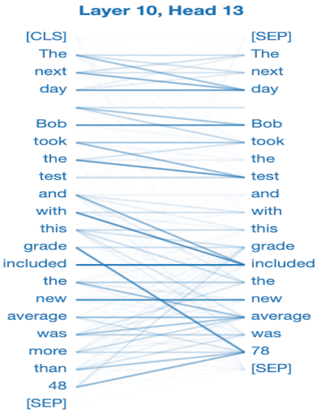

Figure 12: RoBERTa numer. test

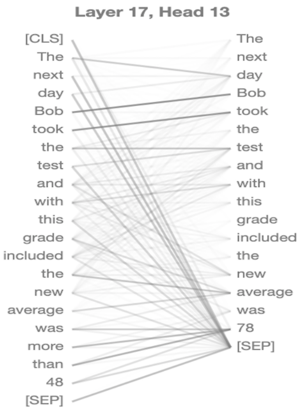

Figure 13: BERT numerical test

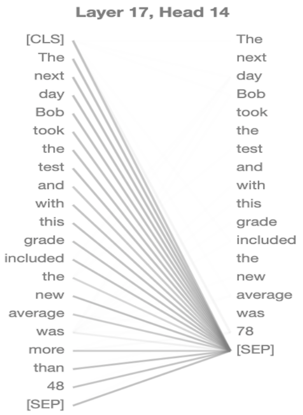

Figure 14: Failed numerical test

Appendix C: Attention-level Results of QA Task

QA task attention-level evaluation

For the QA task, we manually inspected failure cases to see the amount of attention the model paid to the introduced adversaries versus to the correct answer. Here we show one representative example of a *"what"* question:

Question: `What company took over Edison Machine works?.`
Answer: `General Electric.`
Adversary: `Stark Industries took over Calpine Device Well.`

In this particular example, with the question *"What company took over Edison Machine works?"*, the correct answer was *"General Electric"*, and the artificially introduced adversary was *"Stark Industries"*, appended at the end of the context of the original sample.

All models fell into the same trap. It can be seen in Figures 15- 17 that they paid attention to the wrong answer. In this case, this pattern appeared in 52% of the layer-heads of XLNet, 60% in the case of RoBERTa, and 30% on BERT. Nevertheless, while checking the level of certainty of each model in the predicted wrong answer for this example, XLNet had a 43.3% certainty probability, 75.5 % BERT, and the most mistaken was RoBERTa with a 99.9% certainty probability for predicting the wrong answer (which is consistent with the sharpness of attention in Figure 16). This behavior provides evidence that the three models behave slightly different and that increased accuracy in the main task (before adversarial evaluation) is no direct indicator of increased robustness in all cases, but only in the average case.

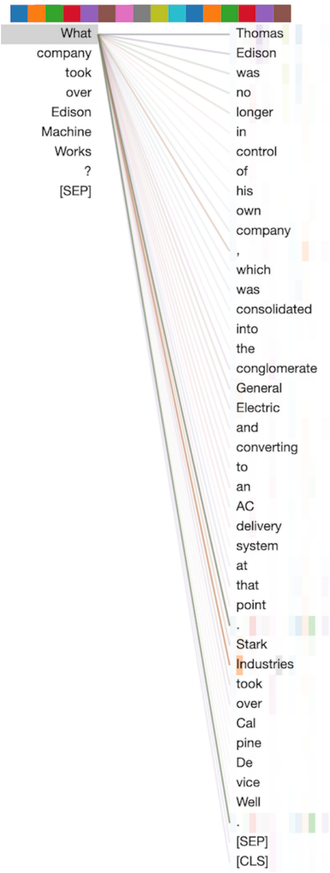

Figure 15: XLNet SQuAD

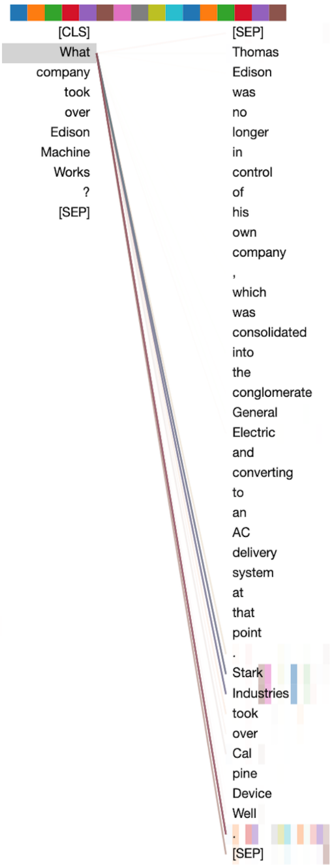

Figure 16: RoBERTa SQuAD

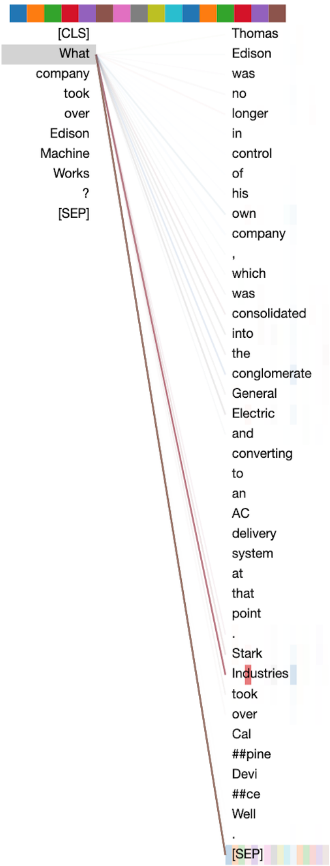

Figure 17: BERT SQuAD