Effect of Ensembling on ANLI Benchmark

Gokhan Cagrici
gcagrici@gmail.com

Abstract

Tremendous achievement of reaching fairly high success metric values with several NLI datasets caused eyebrows to raise questioning the real value of these metric numbers. Research papers started to appear with a comprehensive analysis of what these models really learn and the relative difficulty of forcing these models to fail with small syntactic and semantic changes in the input. In particular, ANLI benchmark is an example of a more challenging NLI task with the intent of measuring the comprehension capabilities of models to a deeper context.

Relative success of transformer-based models on ANLI benchmarks were already reported by Nie et al., 2019. Given the challenging nature of iterative dataset formation, individual models are having more difficulty of extracting the underlying relationship between the context and hypothesis pair, and the target. Ensembles of these individual models might have a higher potential to achieve better performance numbers when the individual performances are that far from the equivalent ones in SNLI and MNLI tasks. On top of that, making controlled variations of the inputs and tracking the changes in the behavior of those models will give indications about the strength and robustness regarding the learning process.

1 Introduction

Models getting closer to human performance numbers for a limited set of input examples but doing very poorly on another are highly subject to suspicion. The idea is not to solely criticize the existing models for not being able to cope with all kinds of input that are substantially different from the training set but to avoid the easiness of tricking a model with little to no effort.

Until recently, recurrent neural networks (RNNs) were seen as the deep learning (DL) recipe to deal with sequences of text. With GRUs and LSTMs, these networks were quite capable of remembering relevant features and forgetting irrelevant ones during the training phase. However, due to the dependency of time step *t* on time step *t - 1*, parallelization became quite challenging. Deep RNNs therefore were really expensive and GPUs were under-utilized with low parallelization factors.

The DL community came up with the idea of transformers a few years ago. Transformers allowed attention mechanisms (similar to the case with RNNs) but without the penalization of being highly sequential. High GPU utilization allowed very deep transformer-based networks, which lifted the bottleneck of being limited to just a few variations of RNNs. Among the popular transformer-based networks available today, BERT gained huge popularity. With this popularity came a lot of variations on the original design as well.

RoBERTa takes the approach of tuning hyperparameters and playing with the training data size. Huge number of parameters involved in BERT design has been an obstacle for many researchers to do parameter searches for their problem domains. RoBERTa is the result of a formal study to analyze the chosen hyperparameters and test the limits of the original design. Training the model longer with more data and longer sequences, changing the static word masking logic with a dynamic one, and removing the next sentence prediction objective let RoBERTa creators achieve better performance numbers on multiple GLUE, SQuAD, and RACE benchmarks. It should be noted that these changes added 15M and 20M new parameters to BERT-BASE and BERT-LARGE, respectively, due to the larger byte-level Byte-Pair Encoding (BPE) vocabulary. (Liu et al., 2019)

XLNet takes a different approach than RoBERTa and criticizes the way BERT was designed. More precisely, the masking approach (autoencoding-

based) during training was replaced with maximizing the likelihood permutations of the factorization order (auto-regressive). While BERT tries to reconstruct the original input from the masked one, XLNet focuses on estimating the probability distribution underlying the input data. Researchers claim that XLNet does not suffer from the unidirectional context typically found in auto-regressive models by depending on all possible permutations of the factorization order, which essentially takes both left and right context into account. Although the next sentence prediction objective was left intact for XLNet-BASE, it was removed for XLNet-LARGE as was the case with RoBERTa because of lack of added value. Apart from these changes, training data size was increased too. The results show consistent improvements on 20 tasks compared to BERT's performance. SOTA results on RACE test data make XLNet the winner too, compared to RoBERTa. (Yang et al., 2020)

With lots of model architectures and specific implementations comes freedom but also confusion about which one to pick. It might also be the case that a model architecture is good at learning a subset of input domain whereas another one is capable of understanding a different subset. Ensemble Learning, which sometimes came to the rescue in these circumstances, is a machine learning technique with the idea of improving the overall performance by combining decisions from multiple networks.

2 Prior Literature

One of the newer adversarial benchmarks to better measure the success of DL models on NLI tasks is called Adversarial NLI with an iterative and human-evaluated framework. With this iterative nature comes the ability to let the benchmark evolve as the community manages to achieve better and better results. The first stage of this benchmark involves human effort to come up with examples on which the current SOTA models are having trouble. These new examples are then used to improve models and a similar procedure is applied to discover more weaknesses. Each round therefore lets new models to be trained while creating more challenging datasets. Diagrammatic explanations of these rounds are shown in Figure 1.

With BERT, XLNet, and RoBERTa chosen as the candidate model architectures, ANLI data were shown to be quite challenging compared to SNLI and MNLI. Even when these models were trained on all three rounds of training datasets (+A1+A2+A3), the test set performance of the best model on ANLI test dataset could not exceed 55.1 accuracy score. Full table of results can be seen in Table 1.

Model	Data	ANLI
BERT	S,F,M,ANLI	49.3
XLNet	S,F,M,ANLI	55.1
RoBERTa	S,F,M,ANLI	53.7

Table 1: Accuracy numbers of BERT, XLNet, and RoBERTa on ANLI. 'S' refers to SNLI, 'M' to MNLI dev, and 'F' to FEVER. (Nie et al., 2019)

Most of the creative ideas about DL architectures in use today for NLP came from the vision domain. Transfer learning, for example, has been in wide use today for vision problems and there are well established pre-trained networks. Besides transfer learning, ensemble learning happens to be another valuable technique for tackling vision challenges. One such a challenging problem of recognizing human actions on still images was tackled with a variety of pre-trained CNNs in isolation and then in an ensembled way. The positive effect of ensembling with varying weighting strategies is shown in Table 2 and Table 3.

Pre-trained CNN	Classification Accuracy
VGG-16	72.61
ResNet50	85.39
InceptionV3	88.81
InceptionResNetV2	89.71
DenseNet201	86.08
Xception	88.83
NASNet-Mobile	85.67
NASNet-Large	91.47

Table 2: Classification accuracies with pre-trained CNNs. (Mohammadi et al., 2020)

Ensemble learning has been used as a valuable technique not only in deep learning but in machine learning as well. At micro level, one can think of a random forest classifier/regressor with an ensemble of decision trees as a strong example.

Given the vision problems have seen significant success so far with ensemble learning, the idea of consulting multiple models for solving NLP tasks would be a logical consequence.

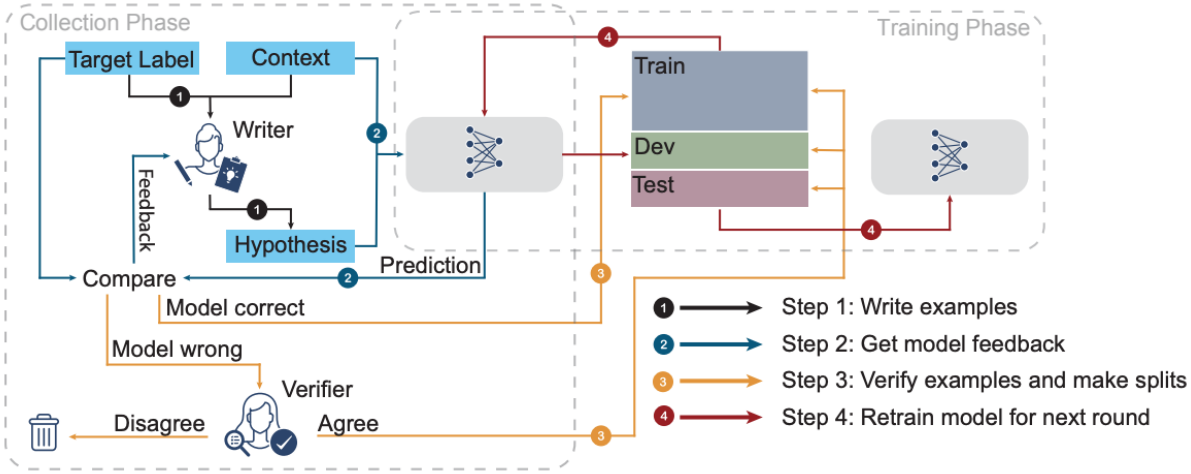

Figure 1: Phases of ANLI dataset creation. (Nie et al., 2019)

Ensemble Method	Classification Accuracy
Averaging on the Best Four Models	72.61
Weighted Averaging on the Best Four Models	85.39

Table 3: Classification accuracies with ensemble methods. (Mohammadi et al., 2020)

3 Data

ANLI dataset formation is a dynamic target for challenging state-of-the-art models to extract the weaknesses of the existing approaches and possibly build better systems. The iterative nature of data collection process with the relatively long context lengths are there for preventing the models for taking shortcuts while reasoning. This iterative nature tries to avoid the shortcomings of a preset target being hit while leaving no room for adjusting the difficulty of inputs. With the help of this calibration process, as models get better and better on the last iteration of the dataset, the moving target is updated with the help of human effort. Educated human annotators as well as several of them checking others' work for correctness help increasing the robustness of data. The official name of this approach is *Human-And-Model-in-the-Loop Entailment Training (HAMLET)*.

First, the procedure starts with a base model trained on the initial training set. Human annotators are then asked to write a hypothesis with respect to a context and target label to force the model to make a classification error. These harder examples are then included in the training set for coming up with a better model. To avoid human mistakes while finding correct set of examples and analyzing the behavior of model predictions, several human verifiers are needed to come to a conclusion about the reason for these model errors. Each of these iterations is called as a round. The original ANLI study stopped iterations after the third round. Completion of these rounds will therefore have three sets of training/dev/test splits. Each round encompasses its own dataset and the model trained within a round is not used in the following rounds. There are three rounds in total. Training data sets are named as A1 (for round 1), A2 (for round 2), and A3 (for round 3). Dataset sizes are shown in Table 4.

The complete dataset grouped by rounds can be downloaded from `https://dl.fbaipublicfiles.com/anli/anli_v0.1.zip`

Sources of data and model architectures per round are described in the following subsections. The focus in this paper will be to use the data points from all rounds instead of progressively training models round by round.

3.1 Round 1

The first round utilizes a BERT-Large model trained on SNLI and MNLI. Wikipedia was chosen as the source of contexts.

3.2 Round 2

The second round depends on a RoBERTa model trained on SNLI, MNLI, FEVER, and the training data from the first round. An ensemble of randomly selected models was chosen to avoid human anno-

tators capture the weakness of a single model and adapt accordingly.

3.3 Round 3

The final round sources data from several domains such as News, fiction, etc in addition to the ones from Wikipedia and the training data from the second round. RoBERTa was chosen as the underlying model one more time.

4 Models

ANLI benchmark used BERT and RoBERTa during the formation of rounds; and BERT, RoBERTa, and XLNet during the evaluation. In this paper, I would like to include all these three model architectures to:

- create BASE and LARGE variants for comparison,
- ensemble best performing models with a weighting strategy,
- analyze how much each one agrees with the ensembled one, and
- make small changes to the correctly classified input examples while testing the robustness of the individual models.

Starting with BERT, a lot of transformer-based networks can consume both a single sequence and a pair of sequences to increase the applicability on multiple tasks. Because NLI task can be formulated as a sequence classification problem, these transformer-based architectures can easily be fine-tuned while feeding `{context, hypothesis}` pairs separated by a special token. This fine-tuning procedure requires special-handling for the outputs, too. It is reasonable to get output as a sequence for a seq-to-seq task, but there are ways to treat part of the output as the predicted class in classification tasks. BERT, for example, suggests using the first token in the output sequence as shown in Figure 2.

By feeding these pair of sequences into the candidate models with true labels, one can expect them to learn the link between the input sequences and between the input pairs and labels. In my opinion, it is a valid argument to think that models pretrained on generic tasks can perform worse than a custom-built model created from scratch for the task at hand but due to the resource and cost limitations, transfer learning is an effective alternative.

One question may arise out of the model selections: Why do I experiment with both BASE and LARGE variants of these model architectures instead of going with the LARGE ones only? It is apparent that research papers usually do not mention about the variants of the networks being presented and that usually means the LARGE variants are being referred. As LARGE variants have roughly three times more parameters than the BASE ones, adding more layers (and therefore more parameters) should increase the capacity with the assumption of suitable model architectures for the task at hand. In this paper, I am trying to analyze if model architectures are applicable to the ANLI task with given data first of all. If the quantitative difference in success metrics for the pair of variants is small, then this might mean that one is not adding huge value by just making the models deeper.

It should be noted that a combined ANLI dataset of all rounds will be fed to those models. On the other hand, Nie et al., 2019 reported results of different models with inputs consisting of not only ANLI dataset but SNLI, MNLI, and FEVER datasets, too. Overall, my experimental setup was briefly shown in Figure 3.

Some details about the model training phase were listed below:

- Models were trained for at least 3 epochs. After the dev dataset performance came to a plateau, training was stopped when the success metric was at maximum.
- Max sequence length setting was set to 150. Despite there were longer sequences, memory requirement of XLNET made it very ineffective to proceed with a higher length.
- Training data were fed to the training phase by random sampling whereas dev/test data were fed by sequential sampling to be able to get the same performance numbers on subsequent evaluations. Random number generator seeds were preset, too.
- BERT and RoBERTa variants were pretrained from the uncased tokens, whereas XLNET was from the cased ones. As a result, preprocessing step for BERT and RoBERTa did the lowercase transformation process for input tokens.
- Context and hypothesis pairs were swapped for RoBERTa model variants.

Dataset	Genre	Context	Train / Dev / Test
A1	Wiki	2,100	16,946 / 1,000 / 1,000
A2	Wiki	2,700	45,360 / 1,000 / 1,000
A3	Various + Wiki	7,000	120,379 / 1,400 / 1,400
ANLI	Various	10,800	162,765 / 2,200 / 2,200

Table 4: ANLI dataset characteristics.

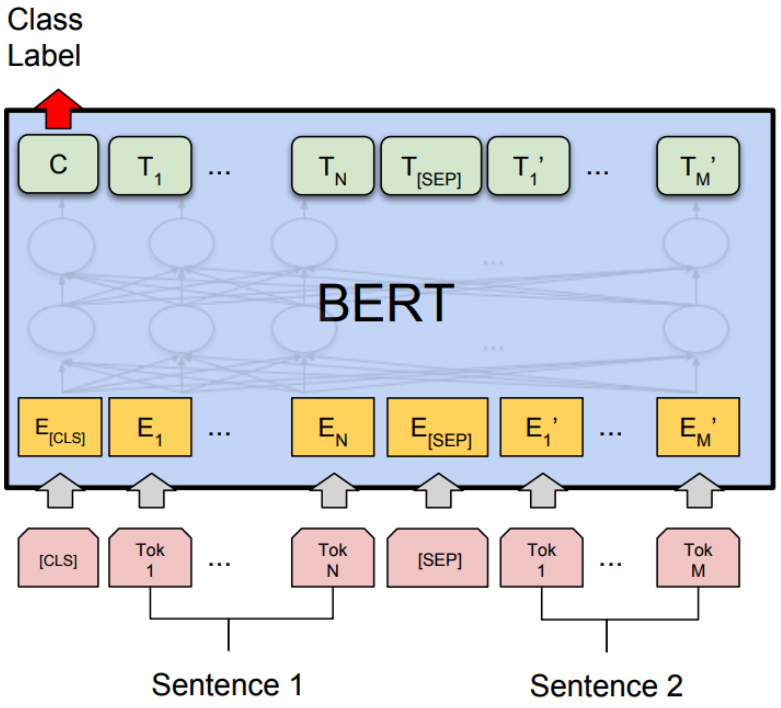

Figure 2: BERT as classifier (Devlin et al., 2019)

- Padding was done on the left side for shorter sequences in XLNET, but on the right in BERT and RoBERTa.
- Dev dataset was only used for hyperparameter search and deciding on where to cut the training process.
- Batch size was set to 8 mainly due to memory restrictions.
- Upon testing with various values, 2e-5 was set as the learning rate for Adam optimizer.
- Distributed training was avoided.

Upon getting the F1 and accuracy numbers per model at the end of training phase, the weighting step was based on multiple strategies. The primary purpose was not only to see if there would be any improvement in the success metrics but also to test if individual models were able to learn different things. Very slim improvements would not have considerable meanings other than occupying a better spot in a leaderboard. Following subsections go over each weighting strategy with examples. Only LARGE variants of the model architectures were included in the weighting study.

4.1 Basic Weighting

Majority of the class predictions per instance will be the decision factor. In case of a tie, the prediction of the best performing model will be chosen. Although this strategy is considering the output of each model equally, one can give more weight to the best performing one.

Probability distributions are not used in this strategy.

For example, if the classes predicted by the three models on instance i are as in (1), the weighted

outcome will be *1*.

$$\begin{aligned} y_{i,1} &= 2 \\ y_{i,2} &= 1 \\ y_{i,3} &= 1 \end{aligned} \tag{1}$$

4.2 Averaging Probabilities Per Class

Probability distributions will be needed in this setup. Probability of belonging to each class as predicted by each model will be averaged and the class with the highest probability will be the answer. In case of a tie, the prediction of the best performing model will be chosen. This strategy is taking the uncertainty into account as opposed to the previous strategy. One can also use the raw activation values prior to softmax.

For example, if the probability distributions of three models on instance *i* are as in (2), the final outcome will be *2*.

$$\begin{aligned} y_{i,1} &= (0.10, 0.80, 0.10) \\ y_{i,2} &= (0.80, 0.05, 0.15) \\ y_{i,3} &= (0.00, 0.11, 0.89) \end{aligned} \tag{2}$$

$$y_{i,avg} = (0.30, 0.32, 0.38)$$

Training a bunch of models, forming an ensemble, and evaluating them on the basis of accuracy and F1 scores is definitely a good start. But, how can we measure the reasoning capacity? My approach will be to find a few test samples that each model could correctly classify and then incrementally complicate the inputs until the models are no longer in consensus. If there happens to surface a pattern for confusion, I can make some statements about the problem points.

5 Results and Analysis

A careful reader may question some of the results to be presented in this section. Following reminders are therefore important:

- ANLI benchmark bases the results on the models trained from ANLI and other datasets such as SNLI, MNLI, etc. This study, however, uses ANLI data only.
- F1 scores are not mentioned in the ANLI benchmark document so both F1 and accuracy numbers are shared in this section.

Three model architectures with two variants produced the accuracy and F1 scores shown in Table 5. These numbers mean the following at first sight:

- LARGE variants are consistently doing better than BASE variants.
- RoBERTa and XLNET were more successful than BERT variants but there is not a clear winner among RoBERTa and XLNET.
- Increasing the number of layers and neurons helped to achieve better success metrics but it does not look like a big achievement. In the example of BERT model, the BASE variant has 110M parameters and the LARGE one has 330M parameters, but the F1 score increased from 46.93 to 48.59. Apparently, just increasing the depth and width of these networks will not be the ultimate solution.
- ANLI benchmark results show slightly better numbers most probably due to the case that additional datasets were used during training and/or more training occurred.

Model	Accuracy	F1
BERT-BASE	47.59	46.93
RoBERTa-BASE	48.16	48.18
XLNET-BASE	48.66	48.38
BERT-LARGE	49.06	48.59
RoBERTa-LARGE	50.66	49.48
XLNET-LARGE	49.16	49.14

Table 5: Experiment Results on Test Dataset

Having seen the results in Table 5, the next reasonable question to ask is if ensembling multiple models can produce better results by combining their capabilities. LARGE variants of these three model architectures were chosen as the top performing ones to be included in the ensemble. Table 6 shares the numbers per the weighting strategy chosen.

Weighting Strategy	Accuracy	F1
Basic Weighting	51.41	50.82
Avg. Prob. per Class	51.63	50.98

Table 6: Ensembling LARGE variants of BERT, RoBERTa, and XLNET

Ensembling the top three performing models via basic weighting strategy helped increasing the best accuracy and F1 numbers so far. This result proves that individual models are better than the others

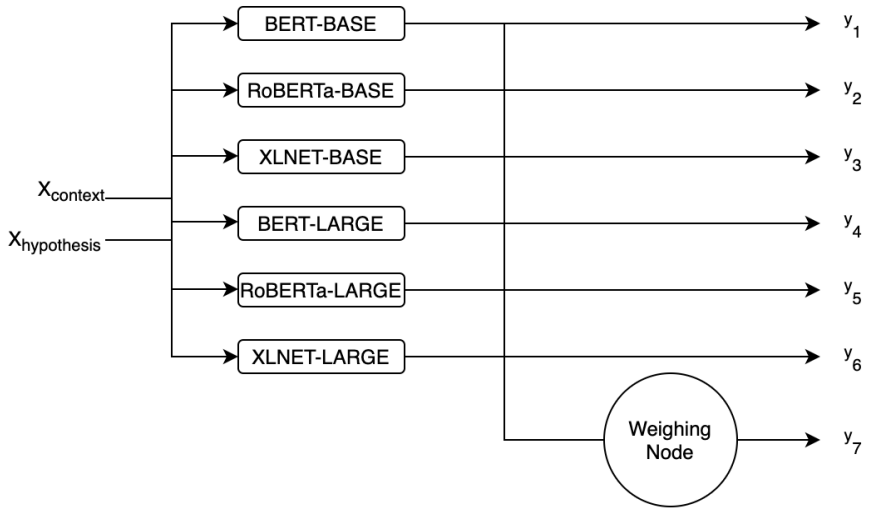

Figure 3: Variants of several transformer models on ANLI with ensembling

Context	Hypothesis	True Label
Trumpkin is a fictional character in C. S. Lewisfantasy novel series "The Chronicles of Narnia". Trumpkin is an intensely practical and skeptical dwarf who lives during the reigns of King Miraz and King Caspian X. He is a major character in "Prince Caspian", briefly mentioned in "The Voyage of the Dawn Treader", and is a minor character in "The Silver Chair".	Trumpkin is an intensely practical and skeptical giant who lives during the reigns of King Miraz and King Caspian X.	c
Even though Congress's pattern has been to treat all authors equally? I mean, the reason that it's been prospective and retrospective is that people should be, people who hold copyrights should be subject to the same regime and not have some people who got their copyrights the week before the law passed treated differently than people who got it the week after.	This is an opinion.	e

Table 7: Simple NLI examples for top three performing models

for certain inputs and ensembling managed to consolidate their strengths. The improvement is not dramatic though and this fact once more validates that the input is really difficult for these model architectures.

Averaging raw probabilities per class let us to take the uncertainty into account. Although basic weighting considers the prediction of two models being the same by just finding the maximum probability per instance, this approach tries to find the class with the strongest belief collected from each model. Indeed, the performance numbers managed to exceed the ones from the basic weighting strategy. This reminds us that blindly trusting predictions solely on the output of softmax function and choosing the one with the largest value might not be the best bet.

To get deeper into the reasoning process of these models, it is beneficial to look at a few easy and difficult examples in the test set.

What are the common characteristics of simple examples in Table 7 and what makes them apart from the hard ones in Table 8? I will try to mention a few standing facts:

- The hypothesis statements in the simple examples are either replicating the words in the context or making general statements. In the first example, all models could notice the difference between the word 'giant' in the hypothesis and the word 'dwarf' in the context. The second example does not actually use a strong fact about the context and makes a very generic statement.
- The instances in the hard category are more demanding in the sense that recognition of cities and ability to count them are required for the first example, whereas capturing the sex from a pronoun is essential for the second example to make a correct prediction. None of the top performing models could make a correct prediction for both of these examples.

Context	Hypothesis	True Label	Preds
Jay Kahn is a Democratic member of the New Hampshire Senate representing the 10th district. The 10 district is located in the southwestern corner of the state and includes Alstead, Chesterfield, Gilsum, Harrisville, Hinsdale, Keene, Marlborough, Roxbury, Sullivan, Surry, Swanzey, Walpole, Westmoreland and Winchester, New Hampshire.	The 10th district includes 14 towns.	c	[e, n, e]
The Baby Allergy. Allie was feeding her baby for the first time. She decided to start with scrambled eggs. Her baby enjoyed the food and ate it quickly.‘. Suddenly he developed a rash and began to cry. She took him to the doctor where she learned he was allergic to eggs.	Allie had a male baby	e	[n, n, n]

Table 8: Hard NLI examples for top three performing models.

Context	Hypothesis	True Label	Preds
Trumpkin is a fictional character in C. S. Lewisfantasy novel series ”The Chronicles of Narnia”. Trumpkin is an intensely practical and skeptical dwarf who lives during the reigns of King Miraz and King Caspian X. He is a major character in ”Prince Caspian”, briefly mentioned in ”The Voyage of the Dawn Treader”, and is a minor character in ”The Silver Chair”.	Trumpkin is not only a dwarf but also lives during the reigns of King Miraz and King Caspian X.	e	[e, n, e]
Even though Congress’s pattern has been to treat all authors equally? I mean, the reason that it’s been prospective and retrospective is that people should be, people who hold copyrights should be subject to the same regime and not have some people who got their copyrights the week before the law passed treated differently than people who got it the week after.	Some may not believe this.	e	[n, n, n]

Table 9: Testing the robustness of easy predictions

Do simple examples let the models learn some valuable insights about the {context, hypothesis} pairs or do models just memorize a map from the pairs to the labels by ignoring most of the content? One way to find out the answer is to make simple changes to the examples in Table 7 and analyze the behavior of the top performing models.

Models proved to stand against simple attacks such as replacing adjectives with their opposites but tricks like replacing a positive statement with double negatives or making generic statements again with negating words were sufficient to deceive at least one of the networks. Table 9 shows these tricks in action for the simple examples from Table 7.

6 Conclusion

Textual data have lots of intricacies like words with different meanings, named entities, negations, coreferences, contexts dispersed in distant locations, slang, syntactic errors, emojis, and so on. Furthermore, rules in one language is all different from another. It has been known that machine learning and especially DL techniques are hungry for data. As the problem domain gets more and more detailed, data size and quality play even more of an important role. With the lack of data, these networks can quickly overfit and devise memoization techniques.

We are living in an era with huge amount of data being collected by substantial number of devices. With this much of data comes the challenge of labeling that for supervised learning techniques. Generic deep learning models helped to relax this prerequisite via transfer learning but the adaptation of a pre-trained model to a new domain is questionable. In this work, I fine-tuned multiple transformer-based networks on the ANLI dataset and analyzed their success metrics independently and as part of a quorum. The behavior of these networks with respect to slightly modified examples was on my radar, too. The results suggest that although the models are robust enough not to be fooled by things like replacement of individual words with their opposites or synonyms, they are not doing well with negations, quantifiers, pronouns, etc, which implies that there is still a long way to go. As a future work, I believe that more data collected with humans being in the loop and trying to train either a pretrained network or a new one from scratch would be valuable to analyze. Unsupervised techniques such as clustering similar context-hypothesis pairs may help increasing the training set size too.

7 References

Y. Nie, A. Williams, E. Dinan, M. Bansal, J. Weston, and D. Kiela. (2019). Adversarial NLI: A New Benchmark for Natural Language Understanding.

J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding.

Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach.

Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. Salakhutdinov, and Q. V. Le. (2020). XLNet: Generalized Autoregressive Pretraining for Language Understanding.

S. Mohammadi, S. G. Majelan, and S. B. Shokouhi. (2020). Ensembles of Deep Neural Networks for Action Recognition in Still Images.