Final Report: Detecting Sarcasm in Reddit Comments

Hoang Ho
College of Information and Computer Science
University of Massachusetts Amherst

Abstract

Language model, pretrained on a large amount of data and further fine-tuned on GLUE tasks, serves as a strong baseline for most NLP tasks. This project aim to test how well language model's knowledge base, semantic understanding and common sense in detecting sarcasm in Reddit Comments. Being able to detect sarcasm is critical in order to correctly understand people's true intention and sentiment. I will investigate different NLP models: (1) ALBERT Sequence Classifier, SOTA in many NLP tasks, fine-tuned on sarcasm classification task (2) contextualized embedding from ALBERT Language Model and enhanced LSTM models, such as, Attentional LSTM and Conditional LSTM. ALBERT, short for A Lite BERT (Bidirectional Encoder Representations from Transformers), (Lan et al., 2020) has never been used for this specific task of sarcasm detection before. I'll show that a simple ALBERT classifier provides surprisingly good results for this task and qualitatively analyze attention from ALBERT and LSTM models to give insight into how well is ALBERT's knowledge base, semantic understanding, and common sense.

1 Introduction

Sarcasm is generally characterized as a figure of speech that involves the substitution of a literal by a figurative meaning, which is usually the opposite of the original literal meaning. Recognizing sarcasm is important for understanding people's actual sentiments and beliefs. For example, a sarcastic sentence ”The U.S is on top of the world in number of CoVid19 cases” can be classified as Positive (In fact, it is classified Positive by Sentiment Analysis demo on AllenNLP). Application of sarcasm detection can benefit many areas of interest of NLP applications, including marketing research, opinion mining and information categorization. However, sarcasm detection is also a very difficult task, as it's largely dependent on context, prior knowledge and the tone in which the sentence was spoken or written. I'll explor how state-of-the-art NLP models make use of context information and attention mechanism in improving classification accuracy and precision.

This project is inspired by Ghosh et al. (2017). In this project, I aim to apply deep learning models with contextualized word embedding such as ALBERT to detect sarcasm in Reddit comments. Experiments with other enhanced LSTM architecture such as Attentional LSTM and Conditional LSTM are performed as a comparison to ALBERT Sequence Classifier. The contributions of the project are: (1) test how well language model's semantic understanding and common sense is; (2) test how well enhanced LSTM architecture with ALBERT contextualized embedding can perform compared to pretrained ALBERT Classifier; (3) verify how well ALBERT Classifier's fine-tuning on GLUE tasks helps performance in totally new task and dataset. In Section 2, I will go over data processing steps for the project. In Section 3, I will go over models details and architectures. Next, I will go over performance for each model, analyzing error and attention weights.

2 Dataset

I will use is Reddit Sarcasm datasset (Khodak et al., 2018) for this dataset. For each examples, there is a parent comment (serve as a context) and a reply (which can be sarcastic or non-sarcastic). The goal is to classify if the reply is sarcastic given the context. Originally, after downsampling and balancing the number of sarcastic and non-sarcastic comments in the original data, I proposed to use a dataset containing a total of 1010826 examples, half of which are sarcastic. However, 1010826 is too big for the computational resource I have, so I have to random sample only 10% of the proposed dataset. The orginal dataset also includes information about authors of the comments, what subreddit the comments belong to and number of down votes and up votes. In this project, I just want to focus on detecting sarcasm from textual information, so information other than the parent comment and reply comment are discarded. The final dataset contains 50542 sarcasm examples and 50542 non-sarcasm examples. The final dataset is split into 75% for training data and 25% for development data.

3 Technical Approaches

In this project, I attempt to perform experiments with several NLP deep learning models and contextualized word embedding techniques. The choice of models are based on how well the model perform on sequence classification tasks, especially Natural Language Inference.

3.1 ALBERT Classifier

My baseline model is ALBERTForSequenceClassification base model. ALBERT (Lan et al., 2020) is a variation of BERT and has significantly fewer parameters than a traditional BERT architecture and can be trained faster than BERT. ALBERT incorporates two parameters reduction techniques: (1) factorized embedding parameterization: decomposing the large vocabulary embedding matrix into two small matrices seperates the size of the hidden layers from the size of the vocabulary embedding; (2) cross-layer parameter sharing: share all parameters across layers. ALBERT is pretrained on masked language model and sentence order prediction (instead of next sentence prediction in BERT). Pretrained model ALBERTForSequenceClassification is fine-tuned for premiliary experiment. ALBERTForSequenceClassification was first pre-trained on masked language model and sentence order prediction and then fine-tuned on GLUE tasks. ALBERT is currently the SOTA in GLUE tasks benchmark. Data is processed in a similar manner as in Next Sentence Prediction/Sentence Order Prediction: [CLS] + $< parent\_comment >$ + [SEP] + $< reply\_comment >$ + [SEP]. Embedding for [CLS] token will be used by ALBERTForSequenceClassification for classification.

3.2 ALBERT + Attentional LSTM

Attentional LSTM (Yang et al., 2016) is an enhanced LSTM model that was shown to perform very well on sequence classification task. The intuition underlying the model is that not all parts of a sequence are equally relevant for answering a query and that determining the relevant sections involves modeling the interactions of the words, not just their presence in isolation. The model includes two levels of attention mechanisms (Bahdanau et al., 2015): one at the word level and one at the sentence level. Attention at word level is used to learn sentence representation, and attention at sentence level is used to learn document representation. This hierarchical architecture lets the model to pay more or less attention to individual words and sentences when constructing the representation of the document. However, due to computational constrain, in this project, I treat comment and reply as a sequence by itself and learn the attention at word level to construct representation for that sequence.

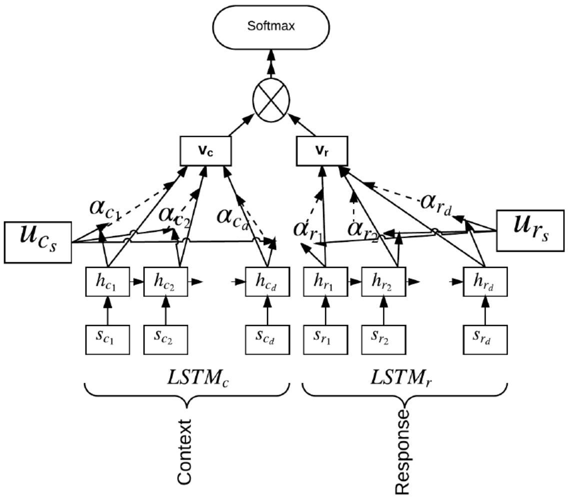


Figure 1: Architecture of Attentional LSTM

I trained jointly 2 different LSTMs: one to learn the representation of the parent comment, and one to learn the representation of the reply. The two representations, v_c and v_r are concatenated and used as input to a softmax output layer.

Using contextualized embedding from ALBERT as input, s_{ct} where c denotes the parent comment, $t \in [1, T]$, and T is the length of the sequence. I use a bidiretional LSTM model to obtain annotations of words $\boldsymbol{h_{ct}} = [\overrightarrow{h_{ct}}, \overleftarrow{h_{ct}}]$. Similarly, for the reply, $\boldsymbol{h_{rt}} = [\overrightarrow{h_{rt}}, \overleftarrow{h_{rt}}]$. Not all words contribute equally to the representation of the sentence meaning. Attention mechanism is used to focus on such words that are important to the meaning of the sentence and aggregate the representation of those words to form an informative representation.

Specifically,

$$\mathrm{u}_{ct} = tanh(W_w h_{ct}) \quad u_{ct} \in R^{2k \times L}$$
$$\alpha_{ct} = \frac{\exp(u_{ct}^T W_{cu})}{\sum_t \exp(u_{ct}^T W_{cu}))} \quad \alpha_{ct} \in R^L$$
$$\mathrm{s}_c = \textstyle\sum_t \alpha_{ct} h_{ct} \quad s_c \in R^{2k}$$

and

$$\mathrm{u}_{rt} = tanh(W_w h_{rt}) \quad u_{rt} \in R^{2k \times L}$$
$$\alpha_{rt} = \frac{\exp(u_{rt}^T W_{ru})}{\sum_t \exp(u_{rt}^T W_{ru}))} \quad \alpha_{rt} \in R^L$$
$$\mathrm{s}_r = \textstyle\sum_t \alpha_{rt} h_{rt} \quad s_r \in R^{2k}$$

Classification:

$$\mathrm{s} = [\mathrm{s}_c, s_r]$$
$$\mathrm{y} = \mathrm{Softmax}(\mathrm{W}_y s)$$

where W_w, W_{cu}, W_{ru}, W_y are learnt parameters, and k denotes the hidden size, L denotes the sequence length. Due to computational constraint, k is chosen to be 128, and max sequence length is 128.

3.3 ALBERT + Conditional LSTM

Conditional LSTM (Rocktäschel et al., 2016) is an enhanced LSTM model specifically designed to learn semantic relationship between a pair sentences, e.g. Natural Language Inference, Recognize Textual Entailment. Rocktäschel et al. (2016) proposed 3 different conditioning techniques for Conditional LSTM, but due to time and resouces constraint, I will only focus on 2 techniques: conditional encoding and conditional attention. Contextual embeddings from ALBERT are input to the LSTM models.

Conditional Encoding model is similar to Attentional LSTM model except that the the final cell state of the parent comment will be used to intialize the cell state of the reply. Thus, the representation for the reply is ”conditioned” on the representation for parent comment.

The intuition for Conditional Attention is that: an LSTM with attention for RTE/NLI does not need to capture the whole semantics of the premise in its cell state; it is sufficient to output vectors while reading the premise and accumulating a representation in the cell state that informs the second LSTM which of the output vectors of the premise it needs to attend over to determine the RTE/NLI class. Let $Y \in R^{2k \times L}$ be a matrix consisting of hidden vectors $[\boldsymbol{h_1}, \ldots, \boldsymbol{h_L}]$ that the first LSTM produced, where k denotes the hidden size. Furthermore, let $e_L \in R^L$ be a vector of 1s and v_r be the representation vector for the reply produced by Attentional BiLSTM as described in the previous subsection. The attention mechanism will produce a vector α of attention weights and a weighted representation $\boldsymbol{r}$:

$$\boldsymbol{M} = tanh(W_y Y + W_r v_r \otimes e_L) \quad M \in R^{2k \times L}$$
$$\alpha = Softmax(W_u M) \quad \alpha \in R^L$$
$$\boldsymbol{r} = Y \alpha^T \quad r \in R^{2k}$$

The final sentence-pair representation is obtained from a non-linear combination of the attention weighted representation of the parent comment $\boldsymbol{r}$ and the reply’s attention weighted representation v_r

$$\boldsymbol{h} = tanh(W_c r + W_x v_r) \quad h \in R^{2k}$$
$$\mathrm{y} = \mathrm{W}_{out} \boldsymbol{h}$$

where $W_y, W_r, W_u, W_c, W_x, W_{out}$ are learnt parameters. Intuitively, the main difference between Conditional Attention LSTM and Attentional LSTM is that Conditional Attention LSTM learns the representation for the parent comment weighted by attention between the representation of the reply and the hidden states of the parent comment, while Attentional LSTM learns the representation for the parent comment weighted by attention for words within the parent comment. Hence, the representation for the parent comment is ”conditioned” on the representation for the reply. Due to computational constraint, k is chosen to be 128, and max sequence length is 128.

4 Experiment

In this section, I’ll go over models’ performance results and perform error analysis and attention analysis to obtain a better understand of model weaknesses.

4.1 Baseline

I fine-tuned pre-trained model ALBERTForSequenceClassification from HuggingFace on detecting sarcasm.

In order to find the optimal hyperparameters to fine-tune ALBERT, I use random search to experiment different combinations of learning rate, batch size and weight decay. Experiments to find hyperparameters are performed on only 10,108 examples. I found that $learning\_rate = 1e - 5$, $weight\_decay = 1e - 3$ and $batch\_size = 32$ gives the best result on data with 10,108 examples. This configuration is used to fine-tuned on the final dataset with 101,084 examples. Early Stopping is used to obtain the best model without overfitting the training set.

We can see that the model has higher recall than precision for detecting sarcasm examples. Even though the dataset is balanced between non-sarcasm and sarcasm examples, the model is biased toward sarcasm examples, i.e. "mistaking" many non-sarcasm examples for being sarcastic. With the same configuration, accuracy increases when data increases 10 times. In Fig. 1, ALBERT overfits the training data after 3 epochs. Given the huge amount of data that ALBERT is pretrained on, a huge amount of data is needed to fine-tune Albert. I hypothesize that if the model is trained on the dataset with 1 million examples, overfitting issue would be less severe and the dev accuracy would be higher. I won't be able to verify this due to computational restriction.

# training examples	Non Sarcasm		
	Precision	Recall	F1
10,108 examples	0.71	0.65	0.68
101,084 examples	0.76	0.68	0.72

# training examples	Sarcasm		
	Precision	Recall	F1
10,108 examples	0.68	0.74	0.71
101,084 examples	0.71	0.79	0.75

# training examples	Dev Accuracy
10,108 examples	71.2%
101,084 examples	74.8%

Table 1: Precision, Recall, F1 Score and Accuracy on Dev Data

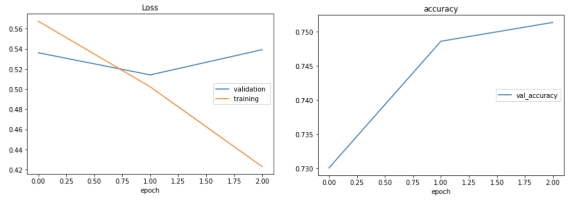


Figure 2: Left: ALBERT Train Loss Dev Loss; Right: ALBERT Dev Accuracy

4.2 LSTM Models

I trained 3 LSTM Models: Attentional LSTM, Conditional Encoding LSTM and Conditional Attention LSTM. All LSTM take ALBERT Contextualized Embedding as input and process the in the manner described in the previous section.

In order to find the optimal hyperparameters, I use random search to experiment different combinations of learning rate, batch size, weight decay, and learning rate schedule. Experiments to find hyperparameters are performed on only 10,108 examples. I found that $learning\_rate = 5e - 5$, $weight\_decay = 1e - 3$, $batch\_size = 32$ and a linear decay learning rate schedule gives the best result on data with 10,108 examples. This configuration is used to fine-tuned on the final dataset with 101,084 examples. Early Stopping is used to obtain the best model without overfitting the training set.

Table 2 shows the precision, recall, F1 score and dev accuracy for three LSTM Models. While Conditional Attention LSTM gives the best result among all three LSTM models, Conditional Encoding LSTM gives similar result to Attentional LSTM. This is surprising because I expected that initializing the cell state of the second LSTM with the final cell state of the first LSTM may give the second LSTM better understanding of the context. All three LSTM models cannot beat ALBERT Classifier, but Conditional Attention LSTM isn't much behind ALBERT Classifier. Given that ALBERT Classifier only has one feed forward layer on top of ALBERT language model and only uses [CLS] token for classifier, superior performance of ALBERT Classifier demonstrates that ALBERT sequence classifier, previous trained on SST-2 and MNLI and WNLI and RTE, generalizes better than just using ALBERT language model. This may be because ALBERT sequence classifier obtains better knowledge base, semantic understanding and common sense through fine-tuning on GLUE tasks.

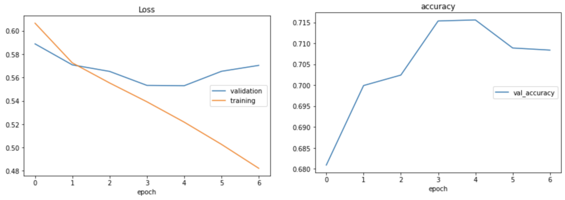


Figure 3: Left: Attentional LSTM Train Loss Dev Loss; Right: Attentional LSTM Dev Accuracy

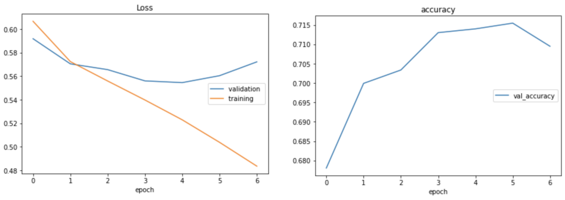


Figure 4: Left: Conditional Encoding LSTM Train Loss Dev Loss, Right: Conditional Encoding LSTM Dev Accuracy

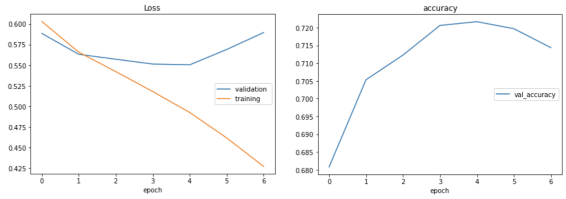


Figure 5: Left: Conditional Attention LSTM Train Loss Dev Loss; Right: Conditional Attention LSTM Dev Accuracy

Model	Non Sarcasm		
	Precision	Recall	F1
Attentional LSTM	0.74	0.65	0.691
Conditional Encoding LSTM	0.734	0.6625	0.6967
Conditional Attention LSTM	**0.731**	**0.685**	**0.707**

Model	Sarcasm		
	Precision	Recall	F1
Attentional LSTM	0.683	0.769	0.723
Conditional Encoding LSTM	0.688	0.757	0.721
Conditional Attention LSTM	**0.70**	**0.744**	**0.721**

Model	Dev Accuracy
Attentional LSTM	71.6%
Conditional Encoding LSTM	71.6%
Conditional Attention LSTM	**72.2%**

Table 2: Precision, Recall, F1 Score and Accuracy on Dev Data

4.3 Error Analysis

I perform error analysis with ALBERT Classifier, the baseline model but also the best-performing model. I randomly sample 512 examples from the validation set and obtain the prediction results for examples. What I observe is that the model seem to be able to understand the meaning of the sentence individually but have some trouble connecting meanings of the pair sentences. Sarcasm usually appears in form of irony, and when the irony doesn't require implication to understand, the model can identify the irony. However, when the irony is subtle requiring domain knowledge or understanding the tone of the conversation or containing double contradictions, the model has some trouble. We may infer from this result that ALBERT Classifiers still lacks some domain knowledge and common-sense to better understand the context and identify the tone of the conversation (see row 8 Table 3). In the next section, I visualize the attention to better understand performance of the model.

Predict	True	Sentence	p(sarcasm)
1	1	"[CLS] popular youtube minecrafter gets exposed for trying to have sex with a child, this is his response video after the numerous accusations against him[SEP] wow, he really seems like a sane individual[SEP]"	0.96
1	1	"[CLS] states project 3 percent increase in prisoners by 2018[SEP] and here i thought crime rates were dropping[SEP]"	0.897
1	0	"[CLS] random but....are those the free ramekins you get with gu stuff?[SEP] forgive my ignorance, what's gu?[SEP]"	0.56
1	0	"[CLS] why is this sub taking my entire childhood and turning it into one big meme[SEP] because your life's a joke[SEP]"	0.786
0	0	"[CLS] making art is my life. check out some of my work and tell me what you think.[SEP] do you have an instagram for your work?[SEP]"	0.13

0	1	”[CLS] what is, in your opinion, the most interesting piece of history you’ve ever learned?[SEP] ireland fought with the axis powers[SEP]”	0.43
0	1	”[CLS] mother’s day parade shooting: at least 12 shot during new orleans festivities[SEP] how could this happen in a state with such lax gun laws?[SEP]”	0.349
0	1	”[CLS] i dunno where the fuck they’re going to / coming from, then. but about 2:30am-3:00am most weeknights, there’s a huge train of people walking by, coming from the direction of the student center and going down the street between buell and best.[SEP] the library maybe?[SEP]”	0.841

Table 3: Examples of correct and wrong predictions and reported probability of the reply being sarcastic

4.4 Attention Analysis

I visualize attention in ALBERT model and Conditional Attention LSTM model.

ALBERT Classifier has 12 attention heads, and after visualizing attention weights from all heads, I observed that in many heads, a word focuses its attention within the surrounding words in the same sentence, and only in a few heads, attention seems to spreads out and words tend to focus on words with similar meanings. I hypothesize that ALBERT learns contextualized representation for words by attending to surrounding words and learns semantic connection between the two sentences by attending to words with similar meanings. More literature reading/review is needed to understand BERT attention heads better. In Fig. 6, words like ”crime”, ”rates”, ”dropping” in the reply spread its attention to words ”prisoners”, ”project”, ”percentage”, ”increase” in the parent comment. However, in second example (Fig. 7), ALBERT Classifier wasn’t able to identify the connection between ”lax”, ”law” in the reply with ”shooting” in the parent comment. Attention visualization further emphasizes that ALBERT Classifier still lacks domain knowledge to make high-level inference between words and pair of sentences.

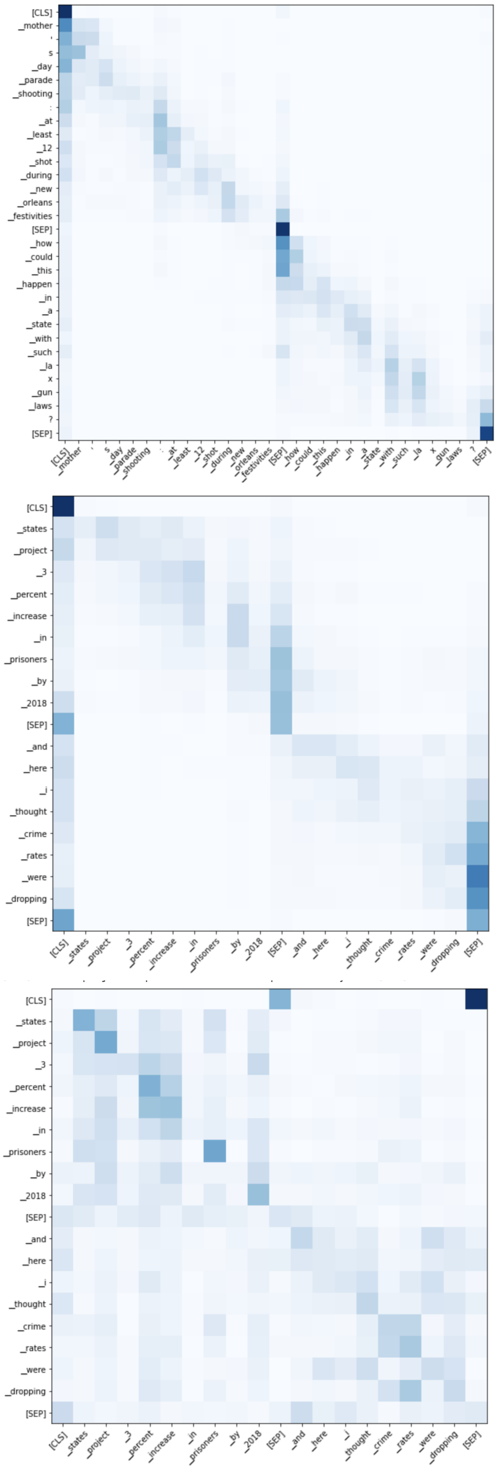


Figure 6: Albert Attention Heads Visualization pred=1 true=1, p(sarcasm)=0.897

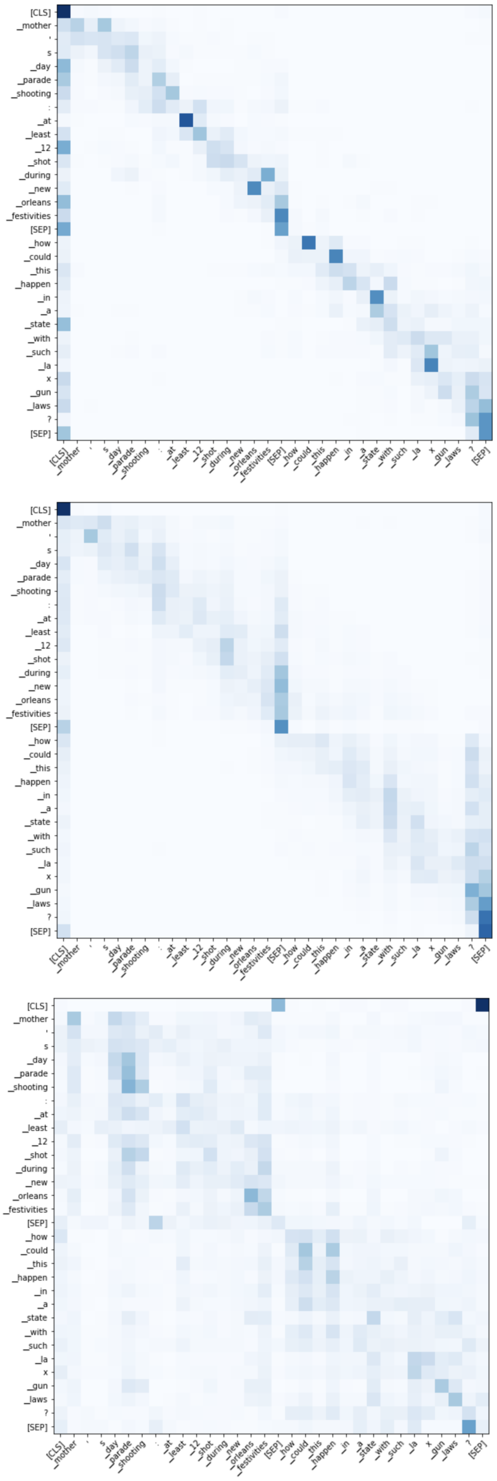
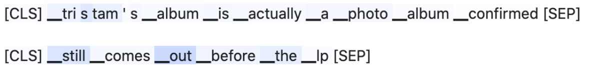


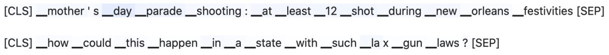
Figure 7: Albert Attention Heads Visualization pred=0 true=1 p(sarcasm) = 0.349

In conditional attention LSTM, there are two types of attention: attention of the reply on the parent comments, and attention on words in the reply itself. In Fig. 8, Fig. 10, Fig. 12, Fig. 13 attention on the parent comment focuses on the words that are most relevant to the context of the reply. Attention on the reply focuses on the words that are most important for classification. In As seen in Fig. 8, attention focuses on words ”still”, ”out”, and in Fig. 9, attention on the parent focuses on words ”at least 12 shot”, and attention on the reply focuses on word ”lax”. In Fig. 12, attention focuses on words ”a wrongful termination lawsuit” from the parent comment and ”shareholders” from the reply, and though these words are important for classification, but the model shortly fails to classify correctly. Similarly in Fig. 13, in the important words ”russia”, ”winter” and ”napoleon”, ”a great” are most attended to, but model fails to identify sarcasm. Examples in Fig. 12 and Fig. 13 require domain knowledge about corporate law and history which the model lacks off.


[CLS] _tri s tam ' s _album _is _actually _a _photo _album _confirmed [SEP]

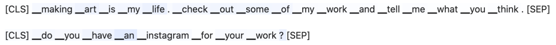
[CLS] _still _comes _out _before _the _lp [SEP]

Figure 8: Conditional LSTM pred=1,true=1, p(sarcasm)=0.86


[CLS] _mother ' s _day _parade _shooting : _at _least _12 _shot _during _new _orleans _festivities [SEP]

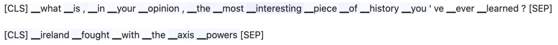
[CLS] _how _could _this _happen _in _a _state _with _such _la x _gun _laws ? [SEP]

Figure 9: Conditional LSTM pred=1, true=1, p(sarcasm)=0.81


[CLS] _making _art _is _my _life . _check _out _some _of _my _work _and _tell _me _what _you _think . [SEP]

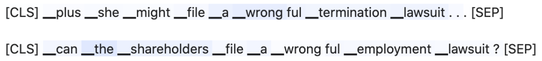
[CLS] _do _you _have _an _instagram _for _your _work ? [SEP]

Figure 10: Conditional LSTM pred=0, true=0, p(sarcasm)=0.205


[CLS] _what _is , _in _your _opinion , _the _most _interesting _piece _of _history _you ' ve _ever _learned ? [SEP]

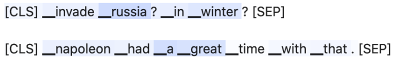
[CLS] _ireland _fought _with _the _axis _powers [SEP]

Figure 11: Conditional LSTM pred=1, true=1, p(sarcasm)=0.64


[CLS] _plus _she _might _file _a _wrong ful _termination _lawsuit . . . [SEP]

[CLS] _can _the _shareholders _file _a _wrong ful _employment _lawsuit ? [SEP]

Figure 12: Conditional LSTM pred=0, true=1, p(sarcasm)=0.453


[CLS] _invade _russia ? _in _winter ? [SEP]

[CLS] _napoleon _had _a _great _time _with _that . [SEP]

Figure 13: Conditional LSTM pred=0, true=1, p(sarcasm)=0.2969

4.5 Hidden Units Ablation Study

Radford et al. (2017) shows that there is a single hidden unit within the multiplicative LSTM that is responsible for sentiment classification results. I attempted to perform an ablation study on Conditional LSTM to investigate hidden units. However, due to the LSTM Models taking so much time to train, I only had a small amount of time working on this ablation study. I investigate the contributions of hidden units on classification results, and I can identify four LSTM hidden units that contribute more than others. However, if I only use these four hidden units for classification, the accuracy on validation set is 63%, and this is very far behind 72.2% accuracy of Conditional Attention LSTM. More research on this topic is still needed.

5 Discussion and Future Work

Although ALBERT is able to generalize very well in detecting sarcasm, ALBERT still lacks some domain knowledge and common sense to make high-level inference. LSTM Models using ALBERT contextualized embedding cannot perform better than a simple ALBERT Sequence Classifier, which implies that ALBERT Sequence Classifier previous fine-tuning on GLUE tasks helps the model to generalize better on a total new dataset. It's often difficult to understand attention in BERT-like model. Future work on interpretability of NLP black box deep learning models will be helpful in understanding the performance and weaknesses of models.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*.

Debanjan Ghosh, Alexander Richard Fabbri, and Smaranda Muresan. 2017. The role of conversation context for sarcasm detection in online interactions. *CoRR*, abs/1707.06226.

Mikhail Khodak, Nikunj Saunshi, and Kiran Vodrahalli. 2018. A large self-annotated corpus for sarcasm. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. European Language Resources Association (ELRA).

Zhenzhong Lan, Mingda Chen, Sebastian Goodman, Kevin Gimpel, Piyush Sharma, and Radu Soricut. 2020. Albert: A lite bert for self-supervised learning of language representations. In *International Conference on Learning Representations*.

Alec Radford, Rafal Józefowicz, and Ilya Sutskever. 2017. Learning to generate reviews and discovering sentiment. *CoRR*, abs/1704.01444.

Tim Rocktäschel, Edward Grefenstette, Karl Moritz Hermann, Tomas Kocisky, and Phil Blunsom. 2016. Reasoning about entailment with neural attention. In *International Conference on Learning Representations (ICLR)*.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489. Association for Computational Linguistics.