# CS 224N Project: Cloze Answer Generator

**Jimmy Zhou**
Department of Computer Science
Stanford University
Stanford, CA 94305
zhouxq@stanford.edu

**Javen Xu**
Department of Computer Science
Stanford University
Stanford, CA 94305
javenxu@stanford.edu

**Jing Bo Yang**
Department of Computer Science
Stanford University
Stanford, CA 94305
jingboy@stanford.edu

## Abstract

Substantial progress has been made in training context-aware language models. The fact that Google's BERT model [1] and Allen Institute's ELMo [2] are occupying majority of SQuAD 2.0 [3] leaderboard reaffirms their effectiveness as an embedding generator. Sentence completion tasks has taken advantage of development in language models, but few have used advanced embedding generation techniques like BERT and ELMo. Moreover, previous works have mostly used single sentences, limiting the amount of context available for a neural language mode. For our work, we build on a freshly developed CLOTH dataset by Xie and Lai [4] which consists of long cloze paragraphs and dozens of blanks in each. Although authors of CLOTH have conducted a number of experiments using *context2vec* [5] and BERT [1], it is still worth investigating the effectiveness other approaches like ELMo, especially with a fine-tuned task specific decoder. Furthermore, we investigate sub-sentence level structures with POS tagging, from which we hope to construct a metric for assessing the difficulty of a cloze paragraph.

## 1 Introduction

Language modeling is the most important component of many natural language processing tasks. Researchers have gone from statistical models that involve building n-gram frequencies to neural models that are increasingly complex. Allen Institute's ELMo[2] and Google's BERT[1] are two impressive context-aware language models that have defeated previous state-of-art methods by large margins. Although they are highly capable models, few have used them for tasks sentence-completion-like tasks. More specifically, previous researchers have only used them for single sentence completion. Our work aims to build a remote-context-aware model based on a freshly collected CLOTH dataset curated by Xie and Lai [4], consisting of multi-blank multi-sentence cloze tests collected from the Internet. During training, cloze samples artificially generated from 1BLM [6] sentences are also introduced to increase the size of training data. We experimented with multiple network structures including modified Position-aware Attention model [7], adapted ELMo, pretrained BERT and custom-made bidirectional-TCN model inspired by the original implementation from [8]. We were able to achieve $53\%$ testing accuracy on an adapted ELMo model and $39\%$ testing accuracy on our custom-made bi-directional TCN model.

In addition to accuracy of selecting an option for cloze tests, we have also analyzed the difficulty of cloze tests from vairous perspectives. We discovered that cloze options generated from using closest

Final Project (2019), CS 224N Natural Language Processing, Stanford.

neighbours of GLoVE [9] are almost 7% easier (absolute improvement) than the original cloze sets. Questions whose options have the same POS tag are more than $10\%$ harder to answer for both ELMo and BERT models.

Besides tackling the cloze test as it is, we also experimented with a different formulation of the problem where we translated the cloze test into an equivalent sentence classification task. In structuring the problem this way, we can evaluate the importance of context in solving cloze test, as sentence classification is essentially the same problem except without the benefit of context in a paragraph. Our task-specific BERT model achieved $88.2\%$ test accuracy on the original cloze test dataset, versus $58.1\%$ accuracy on the re-formulated sentence classification problem, highlighting the importance of context in language models.

## 2 Related Work

### 2.1 ELMo

Released in November 2017, ELMo [2] model is a pre-trained contextual language embedding model which improved the state of the art on several NLP benchmarks. While traditional word embeddings map vocabulary to fixed vectors, ELMo model constructs the embedding by making use of the context in which a word appears. Specifically, ELMo trains a two-layer bidirectional LSTM for language modeling on a large corpus, then use this pre-trained bi-LSTM model as the embedding layer for any model. In this way, ELMo provided a significant step towards pre-training in the context of NLP.

### 2.2 BERT

After ELMo established the utility of pre-trained contextual embeddings in NLP tasks, many architectural advances have happened. In particular, BERT [1] model, developed in 2018, uses Transformers instead of LSTMs in the deep embedding layer. Since BERT's introduction, it has surpassed the performance of ELMo model in a wide variety of benchmarks and has become the new gold standard in many NLP tasks.

A prominent application of the BERT model is for Natural Questions task, where researchers from Google used BERT to establish a new baseline [10]. The authors argued that Natural Questions (NQ) [11] might represent a much harder research challenge than question answering tasks like SQuAD 2.0 and CoQA. Questions in NQ were formulated by people out of curiosity or need for an answer to complete another task *before* they had seen the document that might contain the answer, and the documents in which the answer is to be found are much longer than the documents used in some of the existing question answering challenges. BERT-based performs very well on this dataset, reducing the gap between the model F1 scores reported in the original dataset paper and the human upper bound by $30\%$ and $50\%$ relative for the long and short answer tasks respectively.

In our work, we apply and test both BERT and ELMo models to the cloze test and analyze their performances. Seeing the good performance of BERT model in NQ task, we expect BERT to also have very good performance for cloze test as it shares many similarities with the NQ task. Particularly, the cloze test is also based on long paragraphs for contexts rather than standalone single sentences as in traditional sentence completion task.

### 2.3 TCN

Aside from ELMo and BERT, TCN had been identified to be a suitable replacement for RNN-type structures[8] for tasks including language modeling and music generation. Consisting of dilated causal convolutions, TCN seemed to be a promising structure as it can be trained in parallel and leverage advanced hardware compared to strictly sequential RNN models. Usage of TCN as an embedding generator is shown in Figure 3. Each TCN block in fact has 2 convolutional layers with a residual connection, much like a standard ResNet connection.

# 3 Approach

## 3.1 Baseline Model

We first modified existing preprocessors for the CLOTH dataset [4] because baseline and ELMo [1] models needed additional work on tokenization of the corpus.

We then worked on reproducing the baseline model developed by CLOTH authors. The baseline model's main component was an adapted bidirectional LSTM (*bi-LSTM+Attention* model) with attention score like the Stanford Attentive Reader model [12]. This model is already better than simple biLSTM and unmodified Stanford AR model. We have access to buggy source code of CLOTH authors' implementations [13]. To reproduce their results, we had to use an older version of `pytorch=0.3.1` and fix a variety of things caused by discrepancy between data preprocessing and modeling.

## 3.2 BERT Model

After the introduction of BERT, authors of CLOTH adapted BERT model for the specific task of cloze test. We dissected and reproduced the architecture of the adapted BERT model. See Figure 1 for an illustration of a cloze-specific BERT model.

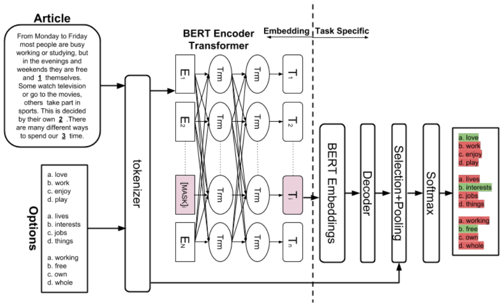

**Figure 1:** Illustration of BERT for cloze architecture implemented by CLOTH authors.

Specifically, the model uses a pre-trained BERT encoder, where each word is masked as a 3-length token. For a regular word such as $home$, the mask vector would simply be $(1, 0, 0)$. For word such as $can't$, the mask vector would be $(1, 1, 1)$, mapping it to token $(can, ', t)$. $512$ is chosen as the maximum number of words in a paragraph prompt for the cloze test. Note that only embeddings of the blank words are extracted for further processing.

The model then uses a decoder with a few fully-connected linear layers which turns the 768-length embedding vector into a 30552-length vector, corresponding to the vocabulary space. Each of the candidate answers are also mapped to a $(3, 30552)$ vector in the output vocabulary space, and compares with blanks from the passage. The last layer is a softmax layer which computes the probability for each of the options, and the model chooses the one with the highest probability.

We were able to reproduced the state-of-the-art result claimed by the authors after fixing many discrepancies between preprocessing and modeling code.

## 3.3 ELMo Model

The next thing we worked on is adaptation of an ELMo model for the same task. This had proven to be difficult because BERT is based on word vectors whereas ELMo uses character-level encoding of words. For this purpose, we modified the preprocessor and the tokenizer for CLOTH to supply input to ELMo.

Specifically, we no longer have to use 3-length tokens. ELMo should be able to handle each option in cloze with one 50-length vector as a character-representation. At the encoder level, ELMo outputs a 1024-length embedding vector. For decoder, we use a few linear layers which turn each character feature vector into 27780-length decoded vectors representing the vocabulary space. Since an embedding-producing ELMo does not need a vocabulary, we had to extract a custom vocabulary list ourselves for our cloze-specific decoder. We then evaluate ELMo using the same method as the BERT model. An illustration of the ELMo architecture is shown in Figure 2. We made use of Allen Institute's ELMo embedding implementation [14], but independently developed the task-specific portion.

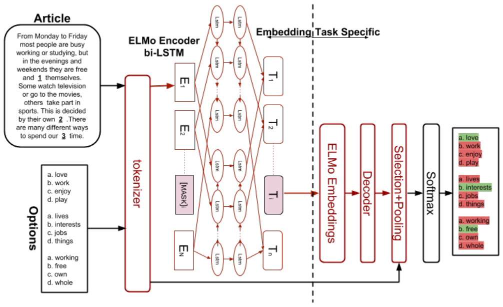
Article
From Monday to Friday most people are busy working or studying, but in the evenings and weekends they are free and 1 themselves. Some watch television or go to the movies, others take part in sports. This is decided by their own 2 .There are many different ways to spend our 3 time.
Options
a. love
b. work
c. enjoy
d. play
a. lives
b. interests
c. jobs
d. things
a. working
b. free
c. own
d. whole
tokenizer
ELMo Encoder
bi-LSTM
E1
E2
[MASK]
EN
Lstm
T1
T2
Ti
Tn
Embedding
Task Specific
ELMo Embeddings
Decoder
Selection+Pooling
Softmax

**Figure 2:** Illustration of an ELMo model adapted for cloze architecture.

### 3.4 Custom Bi-directional TCN

After the re-introduction of Temporal Convolution Network by Bai, Kolter and Koltun [8] to the field of text processing, we believe it was worth trying to replace RNN-structures used by ELMo and Stanford Attentive Reader by TCN. For this purpose, we built a CLOTH processing and learning pipeline from scratch. To emulate the bi-directional nature of common RNN structures, we implemented two parallel TCNs where one of which takes reversed input word vectors. The decoder consists of multiple linear layers. This network is trained exactly the same way as how BERT model is trained - to predict masked tokens. An illustration of this bidirectional TCN network is shown in Figure 3.

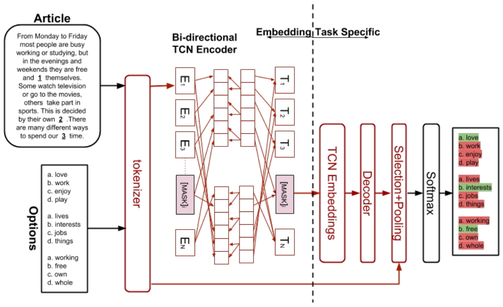
Article
From Monday to Friday most people are busy working or studying, but in the evenings and weekends they are free and 1 themselves. Some watch television or go to the movies, others take part in sports. This is decided by their own 2 .There are many different ways to spend our 3 time.
Options
a. love
b. work
c. enjoy
d. play
a. lives
b. interests
c. jobs
d. things
a. working
b. free
c. own
d. whole
tokenizer
Bi-directional
TCN Encoder
E1
E2
E3
[MASK]
EN
T1
T2
T3
[MASK]
TN
Embedding
Task Specific
TCN Embeddings
Decoder
Selection+Pooling
Softmax

**Figure 3:** Illustration of a bi-directional TCN designed for cloze task

### 3.5 Re-formulation as Sentence Classification Problem

In addition to tackling the cloze test directly, we also experimented with re-formulating the test as sentence classification problems. In approaching it this way, we want to evaluate the importance of context in a cloze test.

We first separate the paragraphs into sentences by punctuation marking full sentences such as question marks, exclamation marks and periods. Then for each blank in a sentence, we fill it with the 4 given choices to make the sentence complete. If a sentence has $n$ blanks, by filling in the choices for each blank, we will end up with $4^n$ sentences. Note that most sentences have just 1 blank, and (in very rare cases) $n$ can go up to 4, which is the maximum number of blanks in a sentence. One of these $4^n$ sentences will be labeled as the correct one and the others incorrect. We then train a task-specific BERT model with these classified sentences.

At prediction, we also fill in blanks of cloze test sentences with the word choices and expand one sentence into a batch of possible choices of complete sentences. We then calculate the probabilities of each sentence from the trained model (with a softmax prediction layer). The sentence with the highest probability is chosen as the final prediction for the correct one and we can retrieve the choices for the cloze test blanks from this sentence.

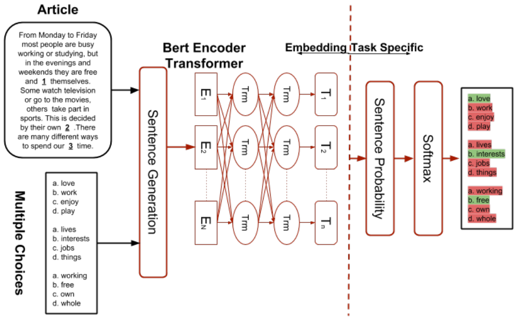

**Figure 4:** Illustration of Bert Sentence Classification for cloze designed for cloze task

# 4 Experiments and Results

## 4.1 CLOTH Dataset

The CLOTH dataset we are using is a protected cloze test dataset which contains 7,131 articles and 99,433 questions. These samples were collected from online resources for Chinese middle-school and high-school English examinations. As described in [4], techniques like OCR and web crawling were used to scrape these materials. Paragraphs with blanks and candidate answers in this dataset were cleaned and stored in JSON format. We gained access to this high quality dataset by contacting the original authors.

There are 4,147 high-school test passages and 3,031 middle-school test passages. We split both high-school and middle-school dataset using a $70 : 15 : 15$ ratio for training, validation and testing.

## 4.2 Artificially Generated Dataset

Although CLOTH dataset already contains plenty of cloze examples, we decided to generate two other datasets to expand our training set size and for more in-depth analysis.

The first is based on CLOTH. We replaced options (3 incorrect options) by nearest neighbours according to GLoVE. This dataset should be more difficult than randomly generated options used by CLOTH authors for their evaluations. Articles and option indices were kept the same.

Another dataset is generated using sentences from 1 Billion World Language Model [6]. We simply concatenated sentences from 1BLM corpus and randomly removed word tokens from the generated article. Again, options were chosen from nearest neighbours on GLoVE. Note that articles in this dataset are not coherent, but a well-trained context-aware model should be able to filter out irrelevant context far from blanks. This method allows us to generated more than 130,000 cloze-like samples and more than 1,600,000 blanks, almost $13\times$ larger than the original CLOTH dataset.

## 4.3 Re-formuated Sentence Classification Dataset

When we fill in the blanks of paragraphs and re-formulate the cloze test as sentence classification problems, we have an transformed dataset of 562,936 training sample sentences. Of these, 64,150 are correct ones and 498,786 are incorrect ones (at least one of the blank is filled with an incorrect word choice). This represents a $7.7 : 1$ ratio between incorrect and correct samples.

## 4.4 Evaluation Method

Evaluation of our models should simply be whether the answer is correct. Thus, we will evaluate percentage-based accuracy of model predictions. Since we have 4 candidate answers for each blank, a random model would have 25% accuracy. Although the TCN model had been trained to recover masked words from the entire vocabulary list (1 in $27k$ base accuracy), it is still evaluated on selecting from 1 out of 4 options the smae way as all other models.

### 4.5 Experiment Details

We ran multiple sets of experiments corresponding to our baseline model, BERT model, ELMo model, custom TCN model and re-formulated sentence classification problem.

We used Azure *NV12* instances with $2 \times M60$ GPUs for large BERT models; AWS *p2.xlarge* with $1 \times K80$ and Azure *NV6* with $1 \times M60$ instances were used for baseline models; another machine with $2 \times GTX1070$ were used for TCN models.

The baseline model is efficient to run. Training each epoch (full run of all training data, no pretraining) takes about 150 seconds. We decay the learning rate from 0.005 to 0.001 and run training for 28 epochs.

For the BERT model, we trained a scaled-down version with 12 embedding layers and a full version with 24 embedding layers. We had access to BERT weights for both models. We trained $4$ samples on each batch, and ran for 2775 batches in total (about 2 epochs). The 12-layer version takes about 1 hours to train. The full version takes much longer, about 5 hours on virtual machine.

For the ELMo model, we only had access to weight for the embedding layer. Training was turned on for ELMo embedding layer when we were training the cloze-specific decoder. Each batch size was 10 samples and learning rate was $1 \times 10^{-4}$. A total of 4 epochs were completed. It is slightly slower than the 12 layer BERT model even when batch size was set to be the same. This might have been cause by inefficiencies related to training long LSTMs.

For custom TCN model, we utilized `pytorch`'s `Dataloader` and `Dataset` class for more efficient data loading. 2 residual blocks were used for each TCN direction and 2 linear layers served as decoder. Thanks to multi-GPU optimization by `pytorch`, we were able to double batch size to 256 and training time for each step was merely 7 seconds.

For the re-formulated sentence classification task, we also used a BERT model with task-specific adaptations from `tensorflow` followed by a softmax classifier.

### 4.6 Results

#### 4.6.1 Cloze Answer Accuracy

The baseline model achieved approximately $53\%$ test accuracy, which is consistent with the original CLOTH author's paper reported result. An off-the-shelf BERT model was able to achieve over $80\%$ best accuracy for prediction among the entire vocabulary list. Once the decoder layers are fine-tuned on cloze dataset, a BERT model can achieve $85\%$ accuracy for high school tests and $88\%$ for middle school tests. These values are similar to those claimed by CLOTH authors. Overall, a fine-tuned BERT model achieves a combined (high school and middle school) accuracy of $86\%$, which is comparable to human performance. An ELMo model without decoder weights was able to achieve $100\%$ training accuracy and $53\%$ testing accuracy. Plots detailing training loss versus accuracy is presented in Figure 5. Various substitute decoder structures were also implemented but none improved significantly compared to the existing transformer-like structure. The custom-made TCN model had the worst performance, only achieving $39\%$ accuracy on the test set, while reaching over $95\%$ training accuracy. We also observed a $58\%$ accuracy on sentence classification method.

| Model | Training Accuracy | Testing Accuracy |
|---|---|---|
| Custom TCN | 95% | 39% |
| LSTM | 82% | 48% |
| Attentive Reader | 95% | 53% |
| BERT Base | 96% | 83% |
| BERT Large | 95% | 86% |
| Adapted ELMo | 100% | 53% |
| Sentence Classification | NA | 58% |

**Table 1:** Performance of tested models

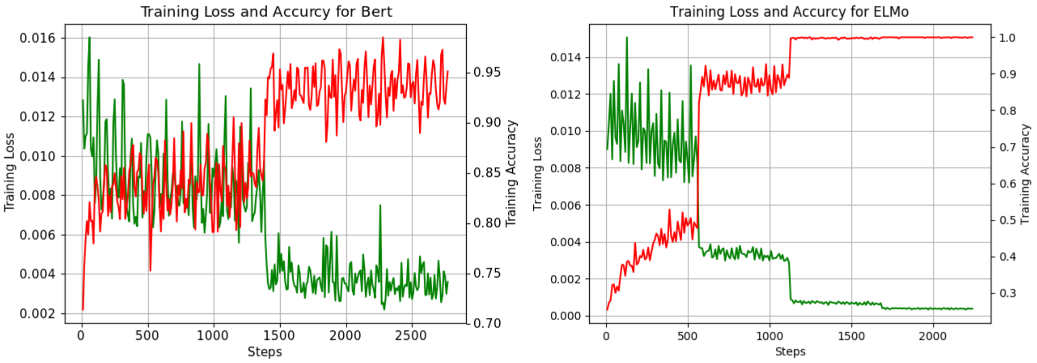

**Figure 5:** Training Loss vs Training Accuracy (Bert on the right, Elmo on the left)

### 4.6.2 Categorized Accuracy

For BERT, accuracy for questions where options are of the same POS tag (for example, all $4$ options are of tag DT (determiner)) is 76%, whereas our ELMo model achieved $40\%$. Both of these are lower than the overall accuracy.

When questions are categorized according to POS tag of the answer, BERT out-performs ELMo on all except `JJS` (adjective, “biggest"). A table detailing this comparison is presented in Table 2. The easiest category are `VBZ` (verb, 3rd person), `WDT`(wh-determiner), `RBR` (adverb) for BERT and are `CC` (coordinating conjunction) and `VBP` (verb, single, present) for ELMo. The most difficult category is `JJR` (adjective, bigger) for both BERT and ELMo. Summary for number of questions belonging to each category can be found in the original CLOTH article [4]. We also have analyzed the categorized accuracy on the classification method, and find out that conjunction categories performed poorly.

| Model | CC | CD | DT | IN | JJ | JJR | JJS | NN | NNS | PRP | PRP$ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bert | 0.78 | 0.5 | 0.82 | 0.76 | 0.71 | 0.4 | 0.45 | 0.67 | 0.70 | 0.87 | 0.81 |
| Elmo | 0.52 | 0.24 | 0.44 | 0.29 | 0.33 | 0.06 | 0.47 | 0.28 | 0.25 | 0.23 | 0.33 |

| Model | RB | RBR | TO | VB | VBD | VBG | VBN | VBP | VBZ | WDT | WP |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Bert | 0.73 | 1.0 | 0.8 | 0.85 | 0.77 | 0.68 | 0.71 | 0.75 | 1.0 | 1.0 | 0.91 |
| Elmo | 0.29 | 0.21 | 0.27 | 0.32 | 0.33 | 0.22 | 0.28 | 0.63 | 0.19 | 0.14 | 0.30 |

**Table 2:** Test accuracy for categorized questions.

### 4.6.3 Accuracy for Generated Dataset

As mentioned before, we replaced current CLOTH options with closest neighbours from GLoVE. Testing accuracy for this dataset using BERT is $92.5\%$. Testing accuracy for ELMo model is $26\%$.

### 4.6.4 Accuracy for Sentence Classification Problem

After we transformed cloze test into batches of completed sentences and modeled it as sentence classification problem, we have 10,718 unique sentences in our test set. Of these, 6,062 are classified correctly by our trained task-specific BERT model. This represents $56.6\%$ accuracy rate.

When we translated these sentences back into cloze test format (i.e. each sentence corresponding to $1$ or more blanks), we have 11,516 unique blanks where the model got 6,696 correct. This represents $58.1\%$ accuracy rate at the blank/option level.

## 5 Analysis

We can attribute high accuracy of BERT to its training method. As discussed before, BERT was trained to recover masked tokens. This is exactly what cloze test is about, albeit using a longer context. In fact, forcing BERT model to select $1$ out of $4$ options only simplified the problem. In contrast, ELMo was trained for next-word prediction, which is not perfectly suited for cloze. The decoder layers have to trained from scratch, making it prone to overfitting, especially compared to the large corpus that BERT was trained on.

Note that training accuracy had been surprisingly high for all tested models, but ELMo, Attentive Reader and TCN all suffered from over-fitting. The attempt to alleviate overfitting by artificially constructing cloze articles yielded little improvement. The artificially constructed articles are perhaps too easy, as suggested by CLOTH authors. The attempt to train on selecting the entire vocabulary but testing on the fixed cloze options was fruitless as well, suggesting fundamental deficiencies associated with these RNN-like structures. Nevertheless, we notice that decreasing learning rate during training improved accuracy, as shown in Figure 5, jumps in accuracy and decrease in loss are likely associated with stepping down learning rates.

There is little surprise that questions with the same POS tag had around $10\%$ lower accuracy compared to the overall score. Intuitively, these questions are harder and can only be selected given semantic meaning instead of from syntax alone. An interesting note is that the absolute advantage of ELMo over BERT on JJS category comes by sacrificing accuracy in JJR. This pattern leads us to believe that ELMo tends to predict the “biggest" when a comparative adjective is needed. In fact, there is potentially difficulty for both language models to identify “bigger" versus “biggest", as JJR category is the most difficult for both models.

It does come with surprise that questions generated using closest neighbours of GLoVE are “easy" to our language models. Closest neighbours computed from GLoVE are supposed to have similar co-occurence, thus fit well into the blank. Clearly, this is not the case, as a $7\%$ improvement in accuracy (for BERT) is observed compared to the original CLOTH dataset. This suggests that cloze designers have better method for identifying misleading answers than those generated from GLoVE neighbours. This confirms that the cloze test challenges students' knowledge on syntax and sentence understanding at the same time.

Lastly, for experiments with the re-formulated sentence classification problem which is equivalent to cloze test, the BERT model with $58.1\%$ accuracy rate performs worse than it does on solving the cloze test directly. We attribute this decline in performance to the lack of context in sentence classification setup.

Context plays an important role especially in sentences with logical connectors. As a simple example, one of the sentence classified wrong is "[Suddenly] Emma stopped and looked ..." where the model chose "[Finally] Emma stopped and looked ...". We can easily see that the model needs more context in order to find the correct connector word.

## 6 Conclusion

The cloze test is an interesting NLP task that has many real life applications, the most notable one being in language tests. We invested several state of the art contextual-based pre-trained language models for cloze test in this paper, using the recently developed CLOTH dataset. Specifically, we recreated the baseline with variation of the Stanford Attentive Reader model [12] and built on top of the BERT [1] model from original author's implementation.

Our main contribution is in implementing the ELMo model and TCN model to tackle the cloze test. Even though they don't achieve results as good as the BERT model, we have provided another example of NLP task where the BERT model is still state of the art.

We also expanded our evaluation metrics to consider POS tags and conducted detailed comparison of BERT and ELMo models within each POS sub-category. We identified a particularly hard category - JJR (adjective, "bigger") which is the most challenging for both models.

Lastly, we found evidence to justify the importance of context in language models and particularly in cloze test. When we re-formulated the cloze test as sentence classification problems, BERT model achieved lower accuracy, without the benefit of context in this setting.

## References

[1] Jacob Devlin et al. "Bert: Pre-training of deep bidirectional transformers for language understanding". In: *arXiv preprint arXiv:1810.04805* (2018).

[2] Matthew E. Peters et al. "Deep contextualized word representations". In: *arXiv e-prints*, arXiv:1802.05365 (Feb. 2018), arXiv:1802.05365. arXiv: `1802.05365 [cs.CL]`.

[3] Pranav Rajpurkar, Robin Jia, and Percy Liang. "Know What You Don't Know: Unanswerable Questions for SQuAD". In: *arXiv preprint arXiv:1806.03822* (2018).

[4] Qizhe Xie et al. "Large-scale Cloze Test Dataset Created by Teachers". In: *arXiv preprint arXiv:1711.03225* (2017).

[5] Oren Melamud, Jacob Goldberger, and Ido Dagan. "context2vec: Learning generic context embedding with bidirectional lstm". In: *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*. 2016, pp. 51–61.

[6] Ciprian Chelba et al. "One Billion Word Benchmark for Measuring Progress in Statistical Language Modeling". In: *arXiv e-prints*, arXiv:1312.3005 (Dec. 2013), arXiv:1312.3005. arXiv: `1312.3005 [cs.CL]`.

[7] Yuhao Zhang et al. "Position-aware attention and supervised data improve slot filling". In: *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. 2017, pp. 35–45.

[8] Shaojie Bai, J Zico Kolter, and Vladlen Koltun. "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling". In: *arXiv preprint arXiv:1803.01271* (2018).

[9] Jeffrey Pennington, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*. 2014, pp. 1532–1543.

[10] Chris Alberti, Kenton Lee, and Michael Collins. "A BERT Baseline for the Natural Questions". In: *CoRR* abs/1901.08634 (2019). arXiv: `1901.08634`. URL: `http://arxiv.org/abs/1901.08634`.

[11] Tom Kwiatkowski et al. "Natural Questions: a Benchmark for Question Answering Research". In: *Transactions of the Association of Computational Linguistics* (2019).

[12] Danqi Chen, Jason Bolton, and Christopher D. Manning. "A Thorough Examination of the CNN/Daily Mail Reading Comprehension Task". In: *arXiv e-prints*, arXiv:1606.02858 (June 2016), arXiv:1606.02858. arXiv: `1606.02858 [cs.CL]`.

[13] Guokun Lai. *BERT CLOTH*. `https://github.com/laiguokun/bert-cloth`. 2018.

[14] Allen Institute. *An open-source NLP research library, built on PyTorch*. `https://github.com/allenai/allennlp`. 2018.