CSE-156 SP24 Analysis Study: RoBERTa for Question Answering

Author - Pramodya Rajapakse
prajapakse@ucsd.edu

1 Introduction

This repository contains my analysis, examples shown, and datasheets for reference.

Task and Model Summary: The NLP task I chose was Question Answering, which is one of the most widely used applications of language models today. Not only is it seen as a benchmark task for evaluating a model's understanding of text, but it's also one of the most practical applications of AI, be it in search engines, smart home devices, and AI assistants like Chat-GPT. Although I originally planned to research the task of Summarization in my proposal, I pivoted to Question Answering because I personally found the topic more intriguing, and wanted to see its current limitations.

The model I chose to evaluate was RoBERTa (facebookresearch), released in the 2019 paper, RoBERTa: A Robustly Optimized BERT Pretraining Approach (4). The developers improved upon Google's BERT model (1) in several ways, including using a byte-level BPE for encoding, dynamic masking of tokens throughout pre-training epochs, training with larger batch sizes, and rigorously tuning hyperparameters. While RoBERTa definitely isn't the newest or largest language model to date, it was still notable for its vast improvements over the original BERT model, especially since it was trained on a significantly larger corpus of text. It was a widely adopted and still effective model, which is why I chose it for this analysis.

Approach and Findings Summary: My approach for analyzing the system was mainly to run the model on a few commonly used Question Answering datasets to act as benchmarks. Using those, I would gain some insight into what types of errors were most common, and from try to determine why these errors come about. The main limitations I identified during my analysis were cases in which the question contained a synonym that the model didn't understand, cases in which there were multiple references to similar words or numbers and sequencing leading to contextual errors, and cases in which a part of the question appeared in the context however the answer was before the phrase which also led to contextual misunderstandings.

2 Your Dataset

Dataset Description: The main two datasets I utilized for this analysis were the SQuAD v1.1 (6) and SQuAD v2 (5) question-answering datasets. The SQuAD v1.1 (6) dataset consists of more than 100k entries crowdsourced from over 500 Wikipedia articles. Each entry consists of a unique string identifier, the title of which article it was extracted from, a short paragraph of context, the question itself, and a set of possible correct answers. The SQuAD v2 (5) dataset uses all the same entries but also includes 50k more adversarial examples, in which the context specifically does not contain the answer to the question. In this case, the model should predict no answer, as it's confidence score should be extremely low for any output. Due to computing constraints, I chose to only run the model on the validation splits of both datasets, which is roughly 10% of the total dataset, however I shuffled the entries to ensure a less skewed distribution of contexts chosen.

Pre-processing Steps: Besides some basic cleanup like removing whitespaces and changing data types for easier evaluation, there was only one major preprocessing step I had to take before I could get a true evaluation of the model. As mentioned before, each entry had a set of possible correct answers, which could be a number, words, or phrases. The most straightforward way of check-

ing if the model's prediction is correct would be to simply check if the prediction is in that set of correct answers. However, if the prediction was a phrase that was perhaps out of order, but still correct (ex. "Court of Justice of the European Union" vs "The European Court of Justice"), I would want that entry to marked as correct. To achieve this, I used a sentence embedding library sentence-transformers, which used a learned encoding from a DistilBERT model. I then got the cosine similarity between the prediction and each of the possible correct answers for that entry, and stored the max similarity score. If the max similarity score was greater than a certain threshold (I used a threshold of 0.7), then the prediction would count as correct. While this method is far from foolproof, as there were some examples that were still falsely marked as incorrect, this method gave a more robust way to obtain entries that are actual failures in terms of language model understanding.

Dataset Choice and What Makes it Challenging: Since the RoBERTa model was trained on the SQuAD v2 (5) dataset, I expected that overall the system would achieve a high accuracy on both datasets. However, this would also make it easier to analyze the error cases, as it would be interesting to see why certain failures still occur even on a system that has been pre-trained and optimized well for this data. The SQuAD v2 (5) dataset's adversarial examples are especially challenging, because the model needs to understand the context text well enough to determine that the given question cannot be answered, and thus needs to give a low output score.

Some basic statistics of both datasets are provided below.

	SQuADv1.1	SQuADv2
size	10570	11873
num of groups	48	35
avg context length	778.98	810.97
adverserial count	0	5945

Here are some examples of an input-output pair for this task:

Example from SQuAD v1.1

```
>> context = "The Panthers
finished the regular season with
a 15-1 record, and...advanced
to their second Super Bowl
appearance since the franchise
was founded in 1995.  The
Broncos...in the Super Bowl."
>> question = "What year was
the Carolina Panthers franchise
founded?"
>> model(context, question)
   "1995"
```

Adverserial Example from SQuAD v2

```
>> context = "In the course of
the...The Duchy of Normandy,
which began in 911 as a fiefdom,
was established by the treaty of
Saint-Clair-sur-Epte between King
Charles III of West Francia and
the famed..."
>> question = "Who established
a treaty with King Charles the
third of France?"
>> model(context, question)
   ""
```

3 Analysis Approach

The method I employed to analyze the system was fairly straightforward. First, I ran the RoBERTa model on both the SQuAD v1.1 (6) and v2 (5) datasets, extracting out the entries that failed. This included entries that the model gave no answer (apart from the adversarial examples in SQuAD v2 (5) in which the question was unanswerable), the entries in which the model gave an incorrect answer (processed through the similarity checker described above), and the entries in which the model gave an answer when it should not have (exclusive to the adversarial examples in SQuAD v2 (5)). From there, I mainly just went through and observed the entries, noting down the most frequent types of errors and patterns between certain failure cases. While I wouldn't be able to get an exact numerical breakdown of the categories of errors, I was able to at least note the most common types of errors, and then try to find patterns between them and hypothesize the causes, which I could later verify through testing. Examples of this testing can be found in the `examples.ipynb` notebook.

I attempted to get a distribution of the failure entries across the various groups to perhaps see if there were artifacts about the input text that were especially difficult for the model to handle. However, the group's performance was heavily dependent on the type of question, so I chose to omit that data in this report (graph can be found in the

squadv1_analysis.ipynb notebook).

4 Errors and their Categorization

Here I will present the most frequent errors I encountered when looking through the failure cases of the datasets, providing examples for each, and discussing conditions in which they appear. Again, these examples can be found in the examples.ipynb notebook. For the sake of space, I will only include the relevant portions of the contexts.

Over-Extraction One very common, and easy to spot error, would be over-extraction, in which the model's prediction would contain the entire question itself, and a large section of the context, most of which is irrelevant to the question. Here is an example:

```
>> context = 'Some theories of
civil disobedience hold that
civil disobedience is only
justified against governmental
entities.  Brownlee argues that
disobedience in opposition to
the decisions of non-governmental
agencies such as...'
>> question = "Who claims that
public companies can also be part
of civil disobedience?"
>> model(context, question)
   "Who claims that public
companies can also be part of
civil disobedience?Some theories
of civil disobedience hold that
civil disobedience is only
justified against governmental
entities.  Brownlee"
```

Ex - 1.1

It seemed like this error generally occurred when the question contained words that were different but semantically meant the same thing, which the model didn't understand. Another example of this error is shown below:

```
>> context = "Terra preta (black
earth), which...The development
of this fertile soil allowed
agriculture and silviculture
in the previously hostile
environment; ..."
>> question = "What did the
development of this fertile soil
provide in hostile environment?"
>> model(context, question)
   "What did the development
of this fertile soil provide
in hostile environment?Terra
preta (black earth), which is
distributed..."
```

Ex - 1.2

Contextual Misunderstanding - Numerical, Sequence, Repeated Words: Another common type of error was contextual misunderstanding, specifically numerically and temporal ambiguity. When it came to contexts that included many number values, especially if they related to the question being asked, the model would oftentimes grab the wrong number. Here is an example:

```
>> context = "This was the first
Super Bowl...Manning and Newton
also set the record for the
largest age difference between
opposing Super Bowl quarterbacks
at 13 years and 48 days (Manning
was 39, Newton was 26)."
>> question = "How much older was
Manning than Newton during Super
Bowl 50?"
>> model(context, question)
   "39"
```

Ex - 2.1

Although the correct answer was "13 years and 48 days", the model lacked the contextual understanding to know which set of numbers to pick given the question's wording.

Another type of contextual misunderstanding is temporal-based, as in the model not being able to maintain an understanding of sequences. The conditions for this type of error to occur are when the context contains several facts about the order in which certain events happened, especially if they are given out of order or in a variety of sentence structures with different words. Here is an example:

```
>> context = "The Yuan dynasty
is considered both a successor to
the Mongol Empire...In official
Chinese histories, the Yuan
dynasty bore the Mandate of
Heaven, following the Song
dynasty and preceding the
Ming dynasty.  The dynasty was
established by Kublai Khan...the
name of the new dynasty as Great
```

```
Yuan and claimed the succession
of former Chinese dynasties..."
>> question = "What dynasty came
after the Yuan?"
>> model(context, question)
   "Song Dynasty"
```

Ex - 2.2

Although the correct answer to this question would be the ”Ming Dynasty”, there were numerous uses of words that indicate a sequence, like ”successor”, ”following”, and ”preceding”, along with the fact that there are many references to different ”dynasty”. Other examples that failed in this way had similar context conditions, with multiple nouns of similar structures being described in some sort of order.

Contextual Misunderstanding - Shared Phrase: The third most frequent type of error I encountered when looking through the failure cases were another type of contextual misunderstanding errors, in which the model would fail to correctly understand the entirety of the given context, and so the specific arrangement of the words would lead to the model predicting the wrong word or phrase. The conditions for this error are typically when the main part of the question appears in a similar form in the context, but the true answer is found in another part of the context. An example is provided below:

```
>> context = "Euglenophytes are
a group of common flagellated
protists that contain
chloroplasts derived from a green
alga.  Euglenophyte chloroplasts
have three membranes..."
>> question = "What kind of
chloroplasts do Euglenophytes
have?"
>> model(context, question)
   "three membranes"
```

Ex - 3.1

The correct answer to this question would be along the lines of ”chloroplasts derived from a green algae”, however the context has a sentence ”Euglenophyte chloroplasts have three membranes”, which is similar in structure to the question, leading to an error. It is often the case that when the context contains a similar phrase to the question, the model will predict the next relevant word or phrase that comes after that common phrase, which may or not be correct. An example of this is shown below.

```
>> context = "Much of...The
principal role of committees
in the Scottish Parliament
is to take evidence from
witnesses, conduct inquiries
and scrutinise legislation.
Committee meetings..."
>> question = "Taking evidence
from witnesses is one of
committees' what?"
>> model(context, question)
   "conduct inquiries and
scrutinise legislation"
```

Ex - 3.2

Although the correct answer in this case should be ”principle role”, the model predicts the words directly after the shared phrase between the question and context ”take evidence from witnesses”, even though the correct answer was before.

5 Discussion

Here I will discuss the reasons why I think each of the exhibited errors occured, and how they can be potentially solved.

Over-Extraction: These cases seem to be mainly semantic errors, in which the model wasn't trained well enough to understand how certain words are synonyms. Through testing, I found that changing the question to more closely align with words found in the context drastically improved performance. For Ex - 1.1, “public companies” and “non-governmental agencies” are essentially synonyms but the model hasn't learned an association between these words. When I replaced these words in the question, more closely aligning it to the context, the model produced the correct answer, without over-extracting more unnecessary text from the context.

```
>> question = "Who claims that
non-governmental agencies
can also be part of civil
disobedience?"
>> model(context, question)
   "Brownlee"
```

This error can even occur for seemingly minor word differences, as shown below with Ex - 1.2 when changing ”provide” to ”allow”, as in the context, the model produces the correct answer.

```
>> question = "What did the
development of this fertile soil
allow in hostile environment?"
>> model(context, question)
   "agriculture and silviculture"
```

This error could be due to a number of reasons, such as certain words appearing infrequently in the training corpus leading to the model not learning the word's meaning well, or an ineffective tokenizer leading to an embedding that doesn't capture enough of the word's semantic meaning.

Contextual Misunderstanding - Numerical, Sequence, Repeated Words: As mentioned above, contextual misunderstandings can come in many different forms. Numerical errors can occur when there are many relevant numbers near the information in the context that the question is referring to. Temporal errors can occur when the model cannot keep track of the ordering of which events occur, especially if they have similar names because then it is hard to distinguish between them. This is likely due to the attention mechanism and long-range dependencies. If these two mechanisms aren't effective, the model will struggle to maintain the correct relationships between certain words, especially if they are similar or across different sentences. Taking positional encoding into account would be especially helpful to distinguish between words so that the model understands the distinction between semantically identical words (like the word "dynasty" in Ex - 2.2).

Contextual Misunderstanding - Shared Phrases: For these contextual errors, I think they can likely be attributed to how the attention mechanism works for this model. If the model notices that a certain phrase or arrangement of words is shared between the question and context, it's likely that the model will give a higher bias toward the words that are nearby or right after that common phrase, even if it isn't the correct answer. If we take Ex - 3.2, we can see there is a common phrase between the question and context, "take(ing) evidence from witnesses". Although the correct answer is "principle role", which came before the phrase, the model probably gives higher attention to the words after, ", conduct inquiries and scrutinise legislation", which is what the model outputs.

Possible Fixes and Existing NLP Literature: Interestingly, I found that many of the reasons why I thought the discussed errors may occur were tackled by Microsoft Research in their 2021 paper, DeBERTa: Decoding-Enhanced BERT with Disentangled Attention (3). The DeBERTa model they developed had 2 key architectural improvements that lead to improved performance over the original BERT model, and the RoBERTa system I analyzed. For one, they implemented a disentangled attention mechanism, essentially splitting the encodings for content and position, and using relative positioning encoding for each token. As I theorized, many of the errors I brought up could be due to an ineffective attention mechanism, which can happen when adding positional and content embeddings together as RoBERTa does, as information about both could be lost. Separating content and positional encodings should at least lessen the degree to which they conflict, and allow tokens to better attend to other relevant tokens.

Since they used relative positional encodings, they also had to employ an enhanced masked decoder, essentially including the absolute positional information after the transformer layers. This is especially important for certain cases like when a word is used multiple times in a sentence but referring to different objects, as the relative position may not be enough. This would likely help in cases like Ex - 2.2 with contextual misunderstanding errors.

6 Conclusion

Recap: In this analysis, I researched the RoBERTa model for the NLP task of Question Answering. I tested this system on both the SQuAD v1.1 (6) and v2 (5) datasets to identify the most common types of errors and what conditions of the context or questions would cause them to appear. Through my testing, there were three frequent categories of errors. When the question contained words that the model didn't recognize as synonyms, the model's doubt would often lead to over-extraction from the context. If there was a significant amount of numbers, terms that signified sequencing, or uses of the same word referring to different objects, there would often be contextual misunderstanding errors. Another kind of contextual error tended to appear when the question and context shared a common arrangement of words, as the model would tend to just output the rest of the sentence after the shared phrase.

Implications of Findings: As I mentioned be-

fore, since the RoBERTa model was trained on the SQuAD v2 (5) dataset, I expected it to achieve a high performance on both in terms of accuracy. In fact, the system did perform well, with 87.1% accuracy on SQuAD v1.1 and 85.6% accuracy on SQuAD v2 (5). However, I was more interested to see which cases would be difficult for a fairly strong model that was well optimized and trained on this data. Beyond Question Answering being seen as a benchmark for evaluating a language models' capacity to understand text, it is also an extremely common use case that will no doubt be even more widespread in the future. So while accuracy can be high, models that are going to be used around the world for potentially important question answering need to be robust. Even though the DeBERTa model was able to improve upon RoBERTa, it's interesting how the system I analyzed, which is a best-case scenario (being asked question specifically on data the model was trained on), there were still numerous errors for seemingly minor changes, like a difference of one word. I think this goes to show that regardless of how impressive modern language models are, the task of reading comprehension itself is far from solved, and there should be more research done and improvements made before we can treat current models as reliable systems for language modeling tasks like Question Answering.

7 Acknowledgements

I did not utilize any Generative AI tools for testing the system nor to complete this report.

References

[1] 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*.

[facebookresearch] facebookresearch. roberta model repository. https://github.com/facebookresearch/fairseq/tree/main/examples/roberta. Accessed: 2024-06-07.

[3] He, P., Liu, X., Gao, J., and Chen, W. (2020). Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

[4] 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*.

[5] Rajpurkar, P., Jia, R., and Liang, P. (2018). Know what you don't know: Unanswerable questions for SQuAD. In Gurevych, I. and Miyao, Y., editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789, Melbourne, Australia. Association for Computational Linguistics.

[6] Rajpurkar, P., Zhang, J., Lopyrev, K., and Liang, P. (2016). SQuAD: 100,000+ questions for machine comprehension of text. In Su, J., Duh, K., and Carreras, X., editors, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.