What errors are made by TyDi QA systems?

Stanford CS224N Custom Project

Nivedita Rahurkar
Department of Computer Science
Stanford University
rahurkar@stanford.edu

Eric Le
Department of Computer Science
Stanford University
leric@stanford.edu

Neeraj Mathur
Department of Computer Science
Stanford University
mathurn@stanford.edu

Abstract

Everyday people turn to Question Answering systems regularly when they are faced with a question and looking for an answer, however sometimes these QA systems answer the questions incorrectly. Our goal is to understand what errors the QA systems trained on Topologically diverse (TyDi) dataset[1] typically makes for e.g. are these errors due to language structure, or how the question is framed or the nature of passage picked to answer the questions? On a high level our findings suggest that some of the causes of the errors were 1.the ambiguity in framing the question, 2. complex structure of the answer passage influences incorrect answers, 3. multiple possible answers. In an attempt to improve the accuracy of the QA systems on TyDi, we built a classifier by fine tuning mbert model.

1 Key Information to include

- Mentor: Matthew Lamm
- Grading Option: 3 Submit for grading
- Contribution by Nivedita: Labels Generation, Training Classifier, Final Report Writing

2 Introduction

Users today are benefited from the QA systems integrated into search engines and digital assistants.Typically, the questions from these users are information-seeking i.e. they want to know the answer to a question they don't know the answer yet. Lately, these Question Answering (QA) systems have emerged as powerful platforms for automatically answering questions asked by humans in natural language generally making it possible to asking questions and receiving the answers using the language an individual is most comfortable with. If a QA system gives an incorrect answer 4 out of 5 times then the users are less likely to turn to the QA system again for seeking answers. So high accuracy is crucial to any QA system. Before we attempt to increase the accuracy, it is important to analyze the error patterns and causes.

3 Related Work

The English-language Natural Questions (NQ) dataset [2] has spur research advances in QA and assisted the research community immensely to for training and evaluating open-domain question answering system. However, there are thousands of different languages in the world, any many of those use very different approaches to construct meaning. Thus, creating machine learning systems that can understand the many ways languages express the meaning is challenging, and training such systems requires examples from the diverse languages of the world.

Google research group has released a question answering corpus - TyDi QA in order to encourage research on multilingual question-answering covering 11 Typologically Diverse languages with over 200K QA pairs. This is the first public large-scale multilingual corpus of information-seeking question-answers pairs using a simple-yet-novel data collection procedure that is model-free and translation free.

Many early QA datasets used by the research community were created by first showing paragraphs to people and then asking them to write questions based on what could be answered from reading the paragraph. However, since people could see the answer while writing each question, this approach yielded questions that often contained the same words as the answer, resulting in ML algorithms trained on such data would favor word matching, oblivious to the more nuanced answers required to satisfy users' needs.

Traditionally the approach used to create multilingual data by translating an English corpus into other languages is vulnerable to introduction of problematic artifacts to the output language such as preserving source-language word order as opposed to flexible word order or using more constrained/formal language as opposed to common native language.

These languages choices are covering variety of disparate language characteristics, for example many languages use non-Latin alphabets, some form words in complex ways, also for some much data is available on the web to very little.

The authors of Tydi expect that the models performing well on the diverse set of languages would generalize across a large number of the world's languages.

There has been some work done for QA system analysis. In [3] the author focus on the importance of question classification to improve answer selection. The paper by Shen et al. [4] talks about the importance of performing sentiment classification by learning matching vectors of each QA pair.

4 Approach

Baseline
Our original baseline system is the open source TyDi QA Gold Passage (GoldP) task[1], which is a simplified version of TyDi QA's primary task. The GoldP task is simplified as follows:

- Only the gold answer passage is provided rather than the entire Wikipedia article
- Unanswerable questions have been discarded
- Evaluated with the SQuAD 1.1 metrics like XQuAD
- Thai and Japanese are removed because of the lack of whitespace

Main Approach
Our approach is divided into the following two phases.

First, we train mBERT [5] jointly on all the languages of the TyDi QA gold passage training data and evaluate on its dev set [1].

- **Evaluation and Comparison:** Once the retraining of mBERT was completed, we evaluated the results using the evaluation script from Bi-directional Attention Flow for Machine Comprehension [6] to measure the performance of the model using "F1" scores and "exact_match" for each language of GoldP task. We compare our F1 scores with the one in the original paper [1] to ensure that results are acceptable and we can use it as our baseline.
- **Labels generation:** In order for us to prepare for classification task(s) we generated labels representing the correct/incorrect answers. We achieved this by modifying the previously discussed evaluation script to generate a dictionary of labels where the key is the question ID and the value is the correct/incorrect label
- **Linguistic Analysis:** We analyzed a sample of both incorrect and correct answers to better understand the linguistic aspect of the problem space. Our hope with this step was to develop a better intuition about the linguistic features that are influencing the erroneous answers given by mBERT QA. We hope to leverage these linguistic features to engineer the set of feature training function that will be used during development of linear and feature-based classifier.

Next, with the labels obtained in the earlier stage we attempted to build a classifier by fine-tuning mBert. whose results would assist in better understanding of linguistic features having influence for incorrect answers. We reviewed the following approaches for this phase.

- **Linear Classifier:** We initially planed to build a simple linear classifier and train this model using the labels generated in the first phase. The intuition behind this task was that, once trained, this classifier would be able to classify the potential incorrect answers. However, with linear classifier we would classify only high level linguistic aspects based on features such as F1 score, precision, recall, lexical overlap or question-answer genre matching.
- **Feature-based classifier:** In the next phase we attempt to do in-depth feature engineering for our classifier by using language features, some examples are listed in **Table 1**.

Table 1: General features for feature-based classifier

Feature	Description
Answer type match	True if the candidate answer contains a phrase with the correct answer type
Pattern match	The identity of a pattern that matches the candidate answer
Matched question keywords	How many question keywords are contained in the candidate answer
Keyword distance	The distance between the candidate answer and query keywords
Novelty factor	True if at least one word in the candidate answer is novel, i.e., not in the question
Punctuation location	True if the candidate answer is immediately followed by a punctuation
Sequence of question terms	The length of the longest sequence of question terms that occurs in the candidate answer

- **Linguistic Analysis:** At this stage, we performed a detailed linguistic analysis on English test set by leveraging various results from both the phases of the project work to help us understand what language features typically causes errors for the QA systems trained on TyDi dataset. For example, are these errors due to certain language structure, or due to how the questions are framed or due to the type of the passage picked to answer the question?

5 Experiments

5.1 Data

The languages of the world exhibit high typological diversity, the World Atlas of Language Structures categorizes over 2600 languages by 192 typological features which includes phenomena such as word order, grammatical meanings, plurality systems and many more. Thus, it's essential to evaluate models on the data that exemplifies this variety. To encourage research on such multilingual QA, the authors released TyDi QA – a QA dataset covering 11 typologically diverse languages with over 200K QA pairs [1]. The authors expect that the models performing well on this diverse set would generalize across a large number of the world's languages and hope to encourage further research for QA models capable of understanding typologically diverse languages.

5.2 Evaluation method

We had 3 stages of evaluation. First, we verify we could achieve near baseline score by retraining the mBert model. Second, with the labels obtained, we train a basic classifier and check its accuracy with the baseline. Third, we hand craft more intricate linguistic features and fine-tune an mBert model and compare the performance.

5.3 Experimental details

To train the baseline mBert, we used the bert base configuration with 12 hidden layers and attention layers and hidden size of 768. We choose the hidden units dropout probability and attention dropout probability as 0.1. The training data set size was 49370 example. The dev set size was 6102 examples. We had to reduce the batch size to 4 due to the GPU capacity limitation.We trained it on the Azure instance with Nvidia's Tesla M60 gpu. The training time was 16 hours.

For the classification task, we generated labels(0/1) for dev set. So our dataset contained 6102 examples with labels. We split the data as 4800 for training, 651 for dev and test each. We used the train batch size of 4 and for prediction and evaluation we used batch size 2. We trained the classifier for 3 epochs. It took about 4 hours. The model architecture for fine tuning mBert included an instance of mbert model. Then we added an Neural Network layer with output size equal to the number of labels (2 in our task). For reducing over-fitting, we added the dropout layer. Finally, we added a softmax layer to give us probabilities for class labels. We calculated cross entropy loss from given labels and predicted probabilities.

5.4 Results

Report the quantitative results that you have found so far. Use a table or plot to compare results and compare against baselines.

- mBert Baseline: **Table 2** illustrates the comparative result analysis of F1 scores using Gold Passage results from the paper [1] and the results from our retraining of the same baseline as described above. These results are as per our expectation as we tried to train the same model with the same dataset as the authors of TyDi QA.

Table 2: F1 Scores - Original from paper and retraining the baseline

	TyDiQA - GoldP (Original)	Baseline retraining
English	67.90	68.09
Arabic	77.20	76.94
Bengali	77.20	76.94
Finnish	71.00	71.75
Indonesian	76.10	76.50
Kiswahili	79.30	80.43
Korean	61.70	62.67
Russian	70.20	70.34
Telugu	82.40	82.30

- Classification task with logistic regression model: **Table 3** shows the results from training a logistic regression task.

Table 3: Linear Classifier Metrics

	Train Set	Dev set
Loss	0.61	0.97
Accuracy	0.81	0.77
F1	0.42	0.39
Precision	0.98	0.92

- Classification with Fine Tuning mBert : **Table 4** shows the results from fine tuning mBert. More graphs for loss, learnt model word embeddings are added in the appendix section.

Table 4: Fine tuned mBert Metrics

	Train Set	Dev set
Loss	0.150	0.38
Accuracy	0.91	0.88
F1	0.85	0.76
Precision	0.95	0.86

6 Analysis

We've analyzed a few sample incorrect answer cases and below are some of the findings:

Example 1 : In Figure 1 the question is not formed correctly. It doesn't say - 'first word spoken by babies'. So this could add some ambiguity. The prediction is the right answer for question - 'first word understood by babies'

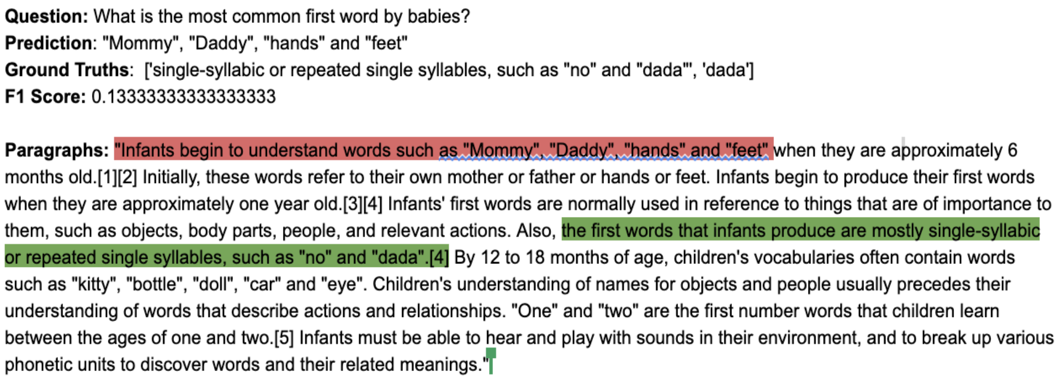
Example 2 : In Figure 2 the network reached conclusion with the first line. It missed the 'previously' and 'However' part.

Example 3 : In Figure 3 it seems like a complex content. It might be hard for the network to understand the embedded text (highlighted in green)

From **Table 3** we can see that linear classifier does not perform well as the features such as F1 score, exact match are high level features while bert has deep word embeddings. From **Table 4** we can see that fine tuning mBert achieves better performance.

7 Conclusion

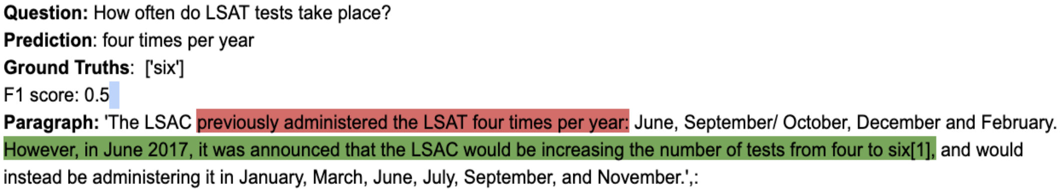
We presented in this paper a method to improve the answering accuracy that first required us to train the mBert on Tydi dataset provided and evaluate the dev set to extract labels for the classification



Question: What is the most common first word by babies?
Prediction: "Mommy", "Daddy", "hands" and "feet"
Ground Truths: ['single-syllabic or repeated single syllables, such as "no" and "dada"', 'dada']
F1 Score: 0.13333333333333333

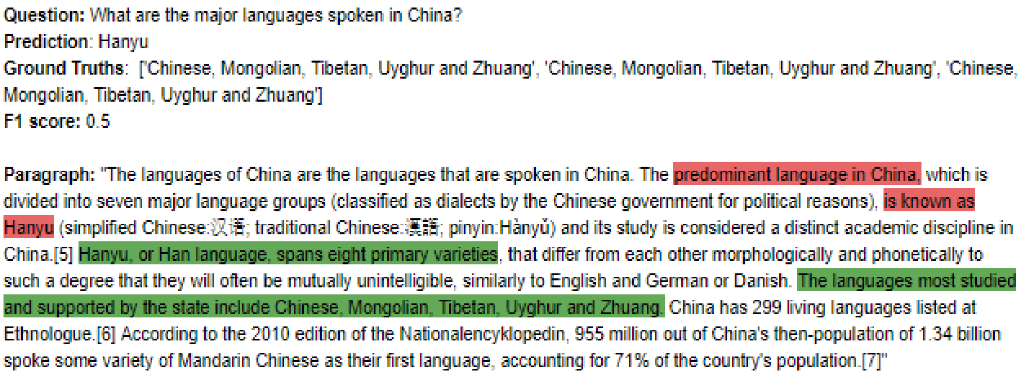
Paragraphs: "Infants begin to understand words such as "Mommy", "Daddy", "hands" and "feet" when they are approximately 6 months old.[1][2] Initially, these words refer to their own mother or father or hands or feet. Infants begin to produce their first words when they are approximately one year old.[3][4] Infants' first words are normally used in reference to things that are of importance to them, such as objects, body parts, people, and relevant actions. Also, the first words that infants produce are mostly single-syllabic or repeated single syllables, such as "no" and "dada".[4] By 12 to 18 months of age, children's vocabularies often contain words such as "kitty", "bottle", "doll", "car" and "eye". Children's understanding of names for objects and people usually precedes their understanding of words that describe actions and relationships. "One" and "two" are the first number words that children learn between the ages of one and two.[5] Infants must be able to hear and play with sounds in their environment, and to break up various phonetic units to discover words and their related meanings."

Figure 1: Error analysis example 1



Question: How often do LSAT tests take place?
Prediction: four times per year
Ground Truths: ['six']
F1 score: 0.5
Paragraph: 'The LSAC previously administered the LSAT four times per year: June, September/ October, December and February. However, in June 2017, it was announced that the LSAC would be increasing the number of tests from four to six[1], and would instead be administering it in January, March, June, July, September, and November.',:

Figure 2: Error analysis example 2



Question: What are the major languages spoken in China?
Prediction: Hanyu
Ground Truths: ['Chinese, Mongolian, Tibetan, Uyghur and Zhuang', 'Chinese, Mongolian, Tibetan, Uyghur and Zhuang', 'Chinese, Mongolian, Tibetan, Uyghur and Zhuang']
F1 score: 0.5

Paragraph: "The languages of China are the languages that are spoken in China. The predominant language in China, which is divided into seven major language groups (classified as dialects by the Chinese government for political reasons), is known as Hanyu (simplified Chinese:汉语; traditional Chinese:漢語; pinyin:Hànyǔ) and its study is considered a distinct academic discipline in China.[5] Hanyu, or Han language, spans eight primary varieties, that differ from each other morphologically and phonetically to such a degree that they will often be mutually unintelligible, similarly to English and German or Danish. The languages most studied and supported by the state include Chinese, Mongolian, Tibetan, Uyghur and Zhuang. China has 299 living languages listed at Ethnologue.[6] According to the 2010 edition of the Nationalencyklopedin, 955 million out of China's then-population of 1.34 billion spoke some variety of Mandarin Chinese as their first language, accounting for 71% of the country's population.[7]"

Figure 3: Error analysis example 3

task. The linear classifier did not perform better than the baseline as Bert model learns more deeper word embeddings while we classified on high level features. The fine tune mBert approach helped to improve the performance of the classification task. Our analysis show that performing classification related to question classification, F1 and exact match scores could provide a strong signal to the QA models and we could improve the current state of the art. So this experiment motivated us to pursue more linguistic features for our task which we plan to do in the future. Since our experiment is limited to Tydi dataset, we would've liked to expand it for other QA datasets. Due to the small labeled dataset and limited resources, it was not possible to finetune on Large Bert model and do hyperparamter tuning.

References

[1] Jonathan H. Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in typologically diverse languages. 2020.

[2] Tom Kwiatkowski and Michael Collins. Natural questions: a new corpus and challenge for question answering research. 2019.

[3] Harish Tayyar Madabushi, Mark Lee, and John Barnden. Integrating question classification and deep learning for improved answer selection. 2018.

[4] Chenlin Shen, Changlong Sun, Jingjing Wang, Yangyang Kang adn Shoushan Li, Xiaozhong Liu, Luo Si, Min Zhang, and Guodong Zhou. Sentiment classification towards question-answering with hierarchical matching network. 2018.

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding.

[6] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. 2016.

A Appendix

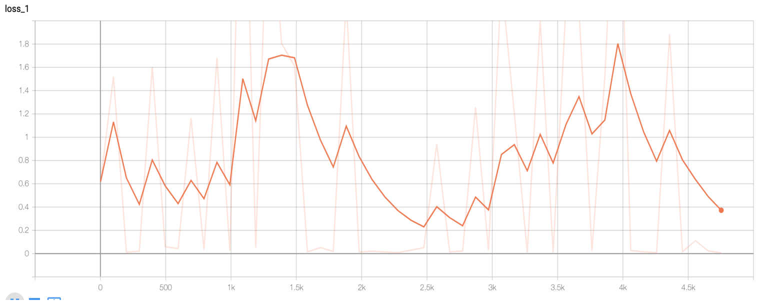


Figure 4: Fine tune mBert: Train Loss

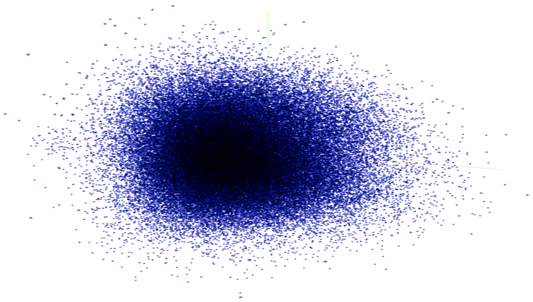


Figure 5: Fine tune mBert: PCA

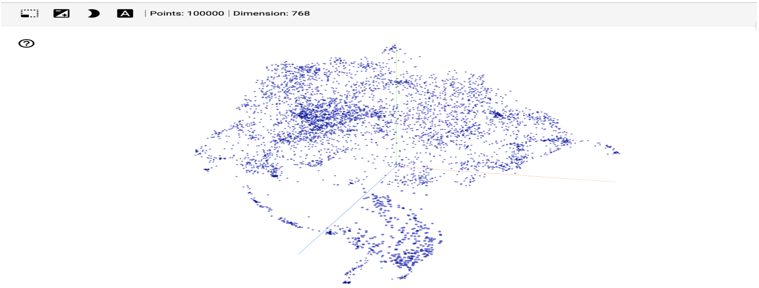


Figure 6: Fine tune mBert: UMAP