Mini-project: Question Word Prediction

Edvard All
edvardal@kth.se

Romain Trost
rtrost@kth.se

May 19, 2022

Contents

1 Introduction

Question answering (QA) is the task of finding some concise, concrete, and correct answer to a given question. This problem is well-studied with both deep learning and knowledge based models having been proposed, and well-performing models have been constructed (see e.g. RoBERTa based CETE by Laskar, Huang, and Hoque [1]).

A different but related problem is question word prediction (QWP), the problem of finding the question word missing from a given question and its corresponding answer. To illustrate, given the following question-answer pair (QAP):

Question <qw> is the capital of Sweden?

Answer: Stockholm

the correct output should be “What“. The constraints on the problem domain are in this sense not obvious: should “Which city“ also be considered a valid question phrase (QP), and so a member of the set of correct answers for above QAP?

2 Background

2.1 Problem constraints

Consider the problem with defining the QWP problem w.r.t. QPs. A question in the English language could be constructed using a number of techniques, such as subject-verb inversion (“Are you tired?“ vis á vis “You are tired.“), but perhaps more commonly the inclusion of question words (QW) e.g. “why“, “what“, “who“, “how“ etc. These QWs are of course central to our problem.

An additional set of QPs could potentially be discerned, loosely following the form “QW followed by some auxiliary determiner“ e.g. “how many“, “which other“, “how far“, “why not“ etc. Yet more QP constructions could be found (e.g. “For what purpose“, “To what end“), and so determining our domain of QPs becomes a problem in itself. To resolve this issue, one can use the constituencies of a sentence to advice QP identification, and importantly what subdomain of QPs that we can include and still have QWP be a tractible problem. This is important not only for QWP itself, but also for how we should construct our dataset – more on this in section 2.3 and 3.1.

2.2 BERT

BERT (Bidirectional Encoder Representations from Transformers) is a machine learning framework that is designed to help computers understand the meaning of ambiguous language in text by using surrounding text to establish context. As its name implies, BERT is based on Transformers, a deep learning model where all input and output elements are interconnected. The use of Transformers allows BERT to read bidirectionally (left-to-right and right-to-left), making it highly competitive in the language model domain.

BERT was pre-trained on the entirety of the English Wikipiedia (unlabeled, plain text corpus), which serves as a base layer of “knowledge“ to build from. From there, BERT can be trained even further through fine-tuning to tackle more specific tasks defined by user’s specifications.

2.3 Stanford Question Answering Database (SQuAD)

SQuAD 2.0 [2] consists of 100’000 questions crowdworked from a set of Wikipedia articles, with corresponding answers derived from the respective articles (if such an answer exists). The set includes 50’000 answerable questions and 50’000 unanswerable questions. It is furthermore important to note that these are all complete questions, and as such cannot be

used (directly) to test our QWP. They further lack POS-tags (which could be used to derive our target QPs as stated in 2.1).

It should here be noted that a subset of questions and interrogative sentences do not conform to our problem, but are nonetheless questions (or in some cases merely interrogative) and so present in SQuAD. A real example of this: ““name a compond that can mobilize sold copper?“. Excluding the fact that the question mark here is non-typical, this sentence is clearly not relevant to our problem, as it lacks a QP. Another real example: “is the cyberspace industry under attack or threatened by regulation? “. Here we have an actual question, but the interrogative nature follows from a subject-verb inversion construction, and so this item in the dataset is also not relevant to our problem.

3 Method

3.1 Constructing the dataset

After some deliberation we settled on a usable, and we argue reasonable, definition of our QPs: “The first question word (determined as such from constituency parsing), including a following adjective of prominent frequence if it is present, of the question“. That we indiscriminately mask the *first* question word, a given sentence may contain several of course: e.g. “What did you do when he left?“, follows from the fact that wh-movement is the norm in the English language, i.e. “When he left, what did you do?“ is a much rarer (although grammatically correct) expression.

This definition has a number of problems which we discuss in section 4. Nonetheless, we accepted these problems based on the limited scope of this project. We can however note that the additional constraint of frequency followed from observation of initially generated QPs, which were so specific that it would be a daunting, and in some cases impossible, task to accurately predict its occurrence in the QP (e.g. “which **arcadian**...“). Based on this definition, we processed the dataset, masking the QPs, and arriving at a set of QPs to be used as input to our predictive model.

3.2 Initial implementation

We initially treated the problem as a simplistic missing word problem, where the location of the missing word(s) is given. An important aspect of this approach was of course that QPs are not always singular, multiple words could constitute a QP. We still hoped to augment such a word prediction model by predicting the probability that some auxiliary determiner followed the most likely question word given the context (question and answer). Then if that probability was too low, we returned the initially predicted QW by its lonesome. This first rudimentary approach proved less than ideal, achieving a prediction score of about 50% on the first 1000 QAPs in the dataset.

Therefore, we decided to treat this problem as a multi-label text classification problem where each label represents a complete QP (e.g: “how much“ is treated as a unique label and not 2 independent words). The aim is for the model to be able to predict the correct label given the masked question along with its answer as input.

3.2.1 Data Sampling

The SQuAD dataset is extremely imbalanced, with the class “what“ making up about 60% of the entire dataset. On the other hand, there are some classes that appear in less than 0.1% of the dataset. Therefore, to ensure that the BERT model could be trained optimally in a way that doesn’t yield poor predictive performance for the minority classes, the majority classes were under-sampled and the extreme minority classes were removed as they contained too little data to train properly (furthermore, these can be considered non-general question words due to their low frequency of occurrence).

In the end, we’re left with a dataset containing 8775 samples made up of 9 evenly distributed classes: “what“, “who“, “when“, “which“, “how many“, “where“, “how“, “why“ and “how much“. These were numerically encoded from 0 to 8 so that the model could handle them efficiently.

3.2.2 Data Preprocessing

The dataset had to be preprocessed into the format expected by the model. To fine-tune the pre-trained BERT model, a tokenizer was built to encode the sentences to ensure that they used the exact same tokenization, vocabulary and index mapping as the BERT model.

The pre-trained model takes 3 equally-shaped tensors as input: (1) `input_word_ids`, (2) `input_mask`, and (3) `input_type_ids`. (1) contains both the tokenized masked question and its tokenized answer concatenated together in the following format: “[CLS] masked question [SEP] answer [SEP]“, where the [CLS] token specifies that this is a classification problem and [SEP] acts as a separator token. (2) is an array containing a 1 anywhere (1) is not padding and 0 otherwise. (3) is an array that contains 0 or 1 indicating if the corresponding token in (1) is part of the masked question or part of the answer.

3.2.3 The Model

In addition to its three inputs described above, the model has a singular output in the form of a one-by-nine array where each value corresponds to the logit for each of the nine classes. In other words, the higher the logit value for a specific class, the higher the chance of that class being predicted as the QW to the given input. The classifier is then trained in a supervised manner (labels are given with each input) using the Adam optimizer with weight decay, which employs a learning rate schedule that firstly warms up from zero and then decays to zero. The model was trained for 10 epochs using a batch size of 32. The predicted label for a given input will correspond to the index of the output array holding the highest value.

3.3 Testing and evaluation

We evalutated our model with respect to our constrained problem, i.e. QAPs where the masked QP is in our restricted QP subset, using train-test split evaluation. We split our initial dataset (as constructed in section 3.1 and **??**) in two, where one part (90%) of the dataset was used to fit the model, and the other 10% were used for this evaluation of the model. The subsets were randomized We consider only predictions which align with what was originally masked as correct predictions.

4 Results and discussion

We achieved an accuracy of 93.7% when evaluating predictions on 878 QAPs. This of course constitutes a major improvement on our initial rudimentary predictive model, which only attained a roughly estimated accuracy of 50%. There is however an important aspect of this evaluation to consider. We evaluate on QAPs that conform to our notion of the problem. Notably, if input questions where the masked QP of which is not in our subset of QPs, the model will inadvertently fail to predict the specific QPs. Therefore this accuracy score could be deemed overly optimistic.

It is important to note however, that this mode of evaluation could simultaneously be argued to be overly pessimistic; In some cases multiple QPs are valid predictions, even in light of the answer. We elaborate this point with a real example from our dataset: {Q; "what singer did beyonce record a song with for the movie, “the best man“?, A; “marc nelson“}. Now let’s mask this question, giving us “<qw> singer did beyonce record a song with /.../?“. It is clear to see that not only “what“, but also “which“, would be a valid answer to this QAP (arguably “which“ is more idiomatic English), yet our evaluation will deem it incorrect. Unfortunately an evaluation that could handle these ambiguous corner-cases would require hand-crafting a data set which was out of scope for this project.

Conclusion

The question word prediction problem is a non-trivial problem, but assuming a limited domain, a predictive model based on text classification using fine-tuned word embeddings works well.

References

[1] Md Tahmid Rahman Laskar, Jimmy Xiangji Huang, and Enamul Hoque. "Contextualized Embeddings based Transformer Encoder for Sentence Similarity Modeling in Answer Selection Task". English. In: *Proceedings of the 12th Language Resources and Evaluation Conference*. Marseille, France: European Language Resources Association, May 2020, pp. 5505–5514. ISBN: 979-10-95546-34-4. URL: https://aclanthology.org/2020.lrec-1.676.

[2] Pranav Rajpurkar, Robin Jia, and Percy Liang. *Know What You Don't Know: Unanswerable Questions for SQuAD*. 2018. DOI: 10.48550/ARXIV.1806.03822. URL: https://arxiv.org/abs/1806.03822.