Two-stage Classification Approach to Automatic Machine Comprehension

Ragheb Al-Ghezi
University of Arizona
Final Project for CSC585
raghebalghezi@email.arizona.edu

Abstract

Machine Comprehension has received great attention from the NLP community not only because solving it will bring us closer to understand humans better, but it also has significant applications in education and automated dialogue platforms. Recently the very task has become even more popular due to the release of SQUAD Competition (Rajpurkar et al., 2018). In this task, a machine learning model is expected to find an answer to a question from a passage, or abstain from answering if an answer is nonexistent. We propose a simple two-stage classification approach for this task. The first stage is training a logistic regression classifier to pick the sentence that is most likely has the answer to the question, or return null if an answer does not exist. After that, we train another logistic regression classifier to choose the right answer span from the best candidate sentence. Our results show this multi-stage approach achieves 0.71 score on the first stage, and 0.%% on the second stage.

1 Introduction

Machine Comprehension (MC) is the task of automatically answering comprehension questions based on information derived from short texts. While this task is relatively easy task for the humans to do, it is a hard task for the machine because it requires not only linguistic information, but also cognitive, world-knowledge skills as well. MC has been of great interest to NLP community because of its practical applications in education, especially in the field of automated assessment and intelligent tutoring systems. Working on similar tasks will bring us a step closer to understand how human is able to perform such cognitive tasks so easily while machine is not.

For this task, we will be using the second release of Stanford Question Answering Dataset (SQuAD) (Rajpurkar et al., 2018). SQuAD 2.0 is a reading comprehension dataset, consisting of 150K questions posed by crowdworkers on a set of Wikipedia articles, where the answer to every question is a segment of text, or span, from the corresponding reading passage, or the question might be unanswerable. The difference between SQuAD 2.0 and its previous release is that it has 50K unanswerable questions in addition to paragraph-supported questions. To do well on SQuAD2.0, systems must not only answer questions when possible, but also determine when no answer is supported by the paragraph and abstain from answering.

The strategy will be used to tackle this task is mainly driven by a simple linguistic intuition. Since SQUAD 2.0 task is an question answering task, meaning the answer span lies within one of given sentences in the paragraph, then an answer is one of the constituents of the best candidate sentence. Therefore, our goal is to, first, find the sentence containing the right answer, and we do by training a logistic regression classifier with some linguistic features. Then, we need another classifier to find the constituent best represent the right answer span within the setence predicted in the first stage. In order to account for the unanswerability of some of the questions, we add a null-answer as a dedicated class in the first classifier along with the potential sentences. This way, if an answer is not found, the inference process stops without proceeding to the next stage, which saves time and computation.

2 Related Work

There has been a large number of studies tackle traditional Machine Comprehension tasks, where answers are extractive and explicitly stated in the

passage. The difficulty of this task, however, lies in abstaining from answers when no answer span exist in the passage. Thus, we will focus on models that not only answers the questions, but also the one that predict the probability of questions being answered.

Seo et al (2016) introduced bi-directional attention flow (BiDAF) that uses an RNN to encode contextual information in both question and passage along with an attention mechanism to align parts of question to the sentence containing the answer and vise versa. The model offers context representation at multilevel of granularity: character-level, word-level and contextual embedding. What sets this work from others is that it does not represent the context paragraph into fixed-length vector. Instead, it dynamically computes vector at each time step, combines it with the one from previous layer and allow *flow* through to the next layers. The model outputs confidence scores of start and end index of all potential answers. One problem with this model is it is not designed to handle unanswerable questions. Levy et al (2017) extends the work by assigning a probability to null-answers to account for the questions whose answers do not exist in the corresponding paragraph. It achieves 59.2% EM score and 62.1% F1 score.

Hu et al (2018) propose a read-then-verify system that is able to abstain from answering when a question has no answer given the passage. They introduce two auxiliary losses to help the neural reader network focus on answer extraction and no-answer detection respectively, and then utilize an answer verifier to validate the legitimacy of the predicted answer. One key contribution is answer-verification. The model incorporates multi-layer transformer decoder to recognize the textual entailment that support the answer in and passage. This model achieves an ExactMatch EM score of 71.6% and 74.23% F1 on SQuAD 2.0.

Wang et al (2018) proposes a new hierarchical attention network that mimics the human process of answering reading comprehension tests. It gradually focuses attention on part of the passage containing the answer to the question. The modal comprises of three layers. a) **encoder layer**: builds representations to both the question and the passage using a concatenation of word-embedding representation (GloVe)@ and a pre-trained neural language modal ELMo @ b) **Attention layer**: its function is to capture the relationship between the question and the passage at multiple levels using self-attention mechanisms. c) **Matching layer**: given refined representation for both question and passage, a bi-linear matching layer detects the best answer span for the question. This method achieves the state-of-the-art results as of September 2018. Their single model achieves 79.2% EM and 86.6% F1 score, while their ensemble model achieves 82.4% EM and 88.6% F1 Score.

While these models achieve astonishing results, they are needlessly complex in terms of architecture design and implementation, and very resource-intensive. This complexity results due the lack of linguistic assumptions that can arguably constrain and direct the problem. The baseline for the task is a simple logistic regression with a set of features, and I would argue that adding more feature could achieve good results at a fracture of cost and complexity.

3 Baseline Implementation

The new version of SQuAD 2.0 task adds a new constraint to competing question-answering models. In addition to identifying the extractive answer spans, a question-answering model should abstain from answering if the passage does not contain an answer. To this end, we implement a simple multinomial logistic regression classifier to address this task. At this level, the classification task is to predict the sentence, in the paragraph, containing the right answer, or declaring that the question is unanswerable. Next, we apply a constituency parser over the sentence predicted from the first stage to get its constituents among which lies the correct answer span (see Figure 1).

3.1 Feature Design

To find the sentence containing the answer, a classifier must determine the sentence that is most similar to the question, and by similar we mean that a good candidate sentence i) shares more words with the question. ii) has high cosine similarity with the question. iii) shares syntactic similarity with the question. Thus, three main features have been selected to this end:

- **Cosine Similarity**: for every sentence in the paragraph as well as the question a word

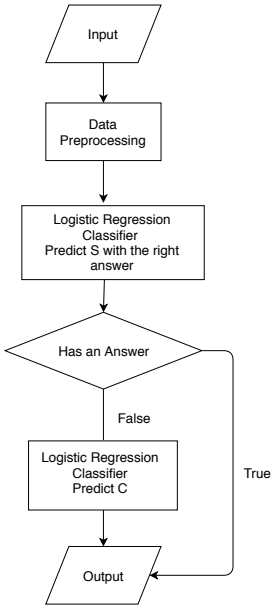

Figure 1: Flowchart illustrating the two-stage classification approach

vector representation is created via InferSent (Conneau et al, 2017), which is a pre-trained sentence embeddings method that provides semantic representations for English sentences. InferSent is an encoder based on a bi-directional LSTM architecture with max pooling, trained on the Stanford Natural Language Inference (SNLI) dataset. Cosine distance score is calculated for each sentence-question pair.

- **Word Overlap**: this calculates the Jaccard score between each sentence-question pair. Jaccard index is a method of computing the explicit similarity between two sets as follows:

$$J(Q, S) = \frac{|Q \cap S|}{|Q \cup S|}$$

where Q and S are sets of words in question and sentence respectively.

- **POS Overlap**: This feature computes the Jaccard score over the part-of-speech-tag representation of sentences. In other words, instead of the word tokens, it checks similarity over POS tokens.We use the default POS-tagger in the Spacy library of Python programming language to obtain the POS representation for the sentences and questions alike.

Using the three features above every question-sentence pair will have three scores and an additional binary feature indicating whether or not the question is answerable. This latter feature is provided by the dataset.

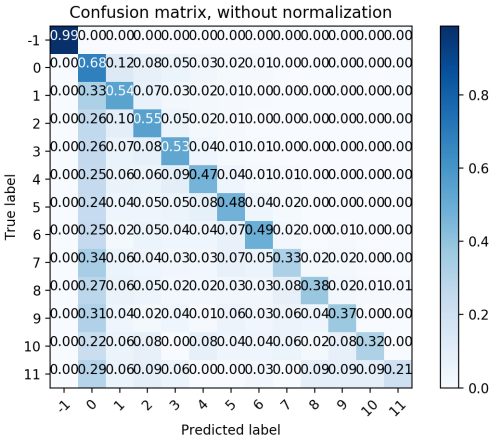

Figure 2: Confusion matrix shows which classes were predicted correctly

3.2 Training and Result

We train a logistic regression classifier with L2 regularization ('newton-cg' solver) using scikit-learn library of Python programming language, and we get the results shown in table 1. Numbers in class column represents the index of the sentence in paragraph containing the answer, and -1 indicates that the question has no answer in the paragraph. We also limit the number of sentence to 10. The results show that with simple features we get an F1 score of 0.71.

class	precision	recall	F1-score
-1	1	0.99	0.99
0	0.47	0.68	0.56
1	0.63	0.54	0.58
2	0.62	0.55	0.58
3	0.62	0.53	0.57
4	0.58	0.47	0.52
5	0.57	0.48	0.52
6	0.57	0.49	0.53
7	0.46	0.33	0.38
8	0.56	0.38	0.45
9	0.46	0.37	0.41
10	0.48	0.32	0.39
11	0.35	0.21	0.26
avg / total	0.72	0.71	0.71

Table 1: This table shows the results of running a multinomial regularized logistic regression. Class column represents the index of sentences within the paragraph, and -1 represent the unanswerable question case. Unpredicted classes are removed.

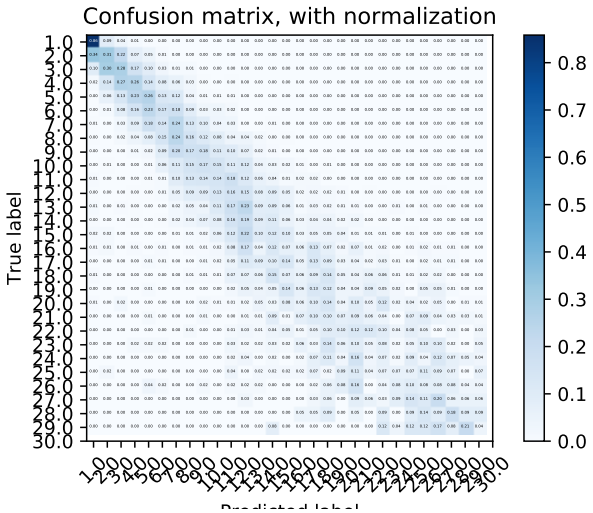

Figure 3: Confusion Matrix illustrating the first iteration of Stage2 LR.

4 Stage Two: Predicting the Answer Span

4.1 Iteration 1

To select the most plausible answer span from the candidate sentence, we design a number of features:

- **Constituents**: Using a constituency parser (Kitaev, 2018), we obtain all constituents in a candidate sentence. These constituents will be the classes from which we pick the right span.
- **Contextual Overlap**: Constituents sharing context with the original question are potential candidates to be the correct answers. So we measure the cosine similarity between each constituent and the question:

$$similarity = \frac{\sum_{i=1}^{n} C_{|w|} Q_i}{\sqrt{\sum_{i=1}^{n} C_{|w|}^2} \sqrt{\sum_{i=1}^{n} Q_i^2}}$$

  where w is the number of slide window around the candidate constituent. For our purposes features of size 2 and 3 are used.
- **Constituent Label**: Constituency parse tree label of the span combined with wh-word.

Out of 85K training example, the answer spans of nearly half of them are not within the constituents. However, answers can be part the constituents. For example, an answer span might be *L'official* and the nearest constituent is *L'official Magazine*. So for our first attempt, we remove all data points whose answers are not explicitly found within the constituents. This results in around 45K data point. Next, we train a logistic regression classifier with L2 regularization and 100 epochs, optimized by newton method 'newton-cg' on a Macbook Pro laptop with core i5 processor and 8 gb of RAM. Python modules used are Scikit-learn and Spacy. The latter is used to drive the attentive-neural constituency parser (Kitaev, 2018). The result is 0.25 F1 score.

4.2 Error Analysis: Iteration 1

By looking at the confusion matrix the first thing we notice is data imbalance. Classes at the beginning have more supporting points, and therefore has less error than the others. However, class 2 has been identified as class 1 34%. For example, the answer of *At what age did Beyonce meet LaTavia Robertson?* is predicted as *At age eight* while the correct answer is *age eight*. Another example, the answer for *How much was public expenditure on the island in 2001-2002?* is predicted to be *Public expenditure* while the true answer is *£10 million*. In this case, the phrase public expenditure appears in the question, which is a stronger candidate given the features designed. Similarly, class 12 is predicted as class eleven 16%. For example, the answer to the question *Who supervised the design*

and implementation of the iPod user interface? is *Steve Jobs*, but it is predicted as *Of Steve Jobs*. Another example, answer to *What roles were women recruited for in the 1950s?* is *in medicine, communication*, but it is predicted as *logistics, and administration*. Given the features we have designed it is very difficult for the classifier to recognize this settle difference between the classes.

4.3 Iteration 2

In the first attempt we have sacrificed valuable data because the answer span was not within the constituents of the candidate sentence. This time, we use all 85K but we will face the same problem introduced in the previous section. The answer span can be a member of more than one constituent in the same sentence, and this will impose a very hard constraint on the performance because the inference can be close but not exact. Also because this problem requires more feature engineering efforts, we decided to leave it for future studies. Instead, we will set the target to the first constituent of which the span is a member. However, we will add three more features:

- **Distributional Distance**: We measure the distributional cosine similarity between the sum of all words in the contextual window and the question using Glove (Pennington, 2012).

- **Matching Word Frequencies**: Sum of the TF-IDF of the words that occur in both the question and the sentence containing the candidate answer.

- **Lengths**: Number of words to the left and to the right of the span.

For classification we compare the performance of a logistic regression classifier with similar configuration as the first stage to 3-layer feed-forward neural network of 128, 64 and 32 nodes respectively. The logistic regression achieves 0.35 F1 while the neural network does 0.42 F1 using Adam optimizer and 100 epochs. Looking at the confusion matrix it is very clear that the new features (tf-idf and distributional cosine similarity) improve the performance. However, they could not recognize the settle difference among the phrase.

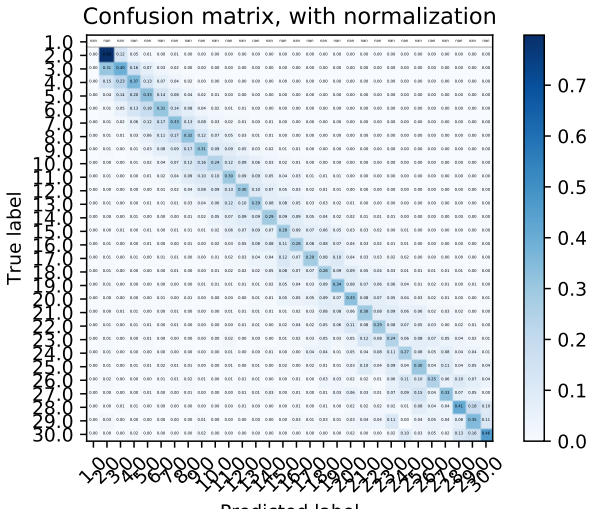

Figure 4: Confusion Matrix illustrating the Second iteration of Stage2 LR.

5 Conclusion

We introduced a two-step classification method for automatic reading comprehension via SQUAD 2.0 dataset. Our stage1 classifier managed to find whether or not a question is answerable within a given passage and find the sentence containing the right answer with F1 score of 0.71. Our stage2 classifier manages to detect the exact span with F1 score of 0.35 even though the predicted answer is not distant from the exact answer. In order to improve the performance of our approach, future studies should investigate the usefulness of features generated from Named Entity Recognition, Semantic Role Labeling and Dependency Parsing processes, which are expected to be potential solutions to the problems we faced in this work.

References

[1] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, and Antoine Bordes. Supervised Learning of Universal Sentence Representations from Natural Language Inference Data. May 2017.

[2] Minghao Hu, Furu Wei, Yuxing Peng, Zhen Huang, Nan Yang, and Ming Zhou. Read + Verify: Machine Reading Comprehension with Unanswerable Questions. *arXiv:1808.05759 [cs]*, August 2018. arXiv: 1808.05759.

[3] Omer Levy, Minjoon Seo, Eunsol Choi, and Luke Zettlemoyer. Zero-Shot Relation Extraction via Reading Comprehension. *arXiv:1706.04115 [cs]*, June 2017. arXiv: 1706.04115.

[4] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ Questions for Machine Comprehension of Text. *arXiv:1606.05250 [cs]*, June 2016. arXiv: 1606.05250.

[5] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional Attention Flow for Machine Comprehension. *arXiv:1611.01603 [cs]*, November 2016. arXiv: 1611.01603.

[6] Wei Wang, Ming Yan, and Chen Wu. Multi-Granularity Hierarchical Attention Fusion Networks for Reading Comprehension and Question Answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1705–1714, Melbourne, Australia, July 2018. Association for Computational Linguistics.