CITS4012 Project

Group 28: Max Chatfield and Nicholas Clarke

University of Western Australia, Australia  
[2](mailto:caren.han@uwa.edu.au)2176321@student.uwa.edu.au

10417777@student.uwa.edu.au

1. Dataset

We used a predefined split of training and test data from the WikiQA dataset. This dataset was assembled from Bing query logs from May 1st 2010 to July 31st 2011 and the summary sections of Wikipedia pages linked to these queries [1]. In this data, candidate answers to the question were labeled 0 and 1, with 1 indicating that the sentence was judged an answer to the question. There were some queries for which there were multiple, or no, sentences which were labeled as answering the question.

The dataset consisted of one row for each sentence of the document, we extracted the question sentence, and collated all sentences associated with that question id along with their answer labels.

Some of the text had irregular spacing, which caused issues for our embedding functions, so we converted all multiple whitespaces to single spaces through a simple regex substitution. We also used this step to space out punctuation from adjacent words.

We chose to keep most punctuation symbols in the data, because we felt that things like brackets were necessary for the context and meaning of certain sentences, and full stops would help demarcate separation of sentences. We also noticed that there was a question “what is the @ sign called?” which implied that for some questions in the dataset certain non-alphanumeric characters might play an important role in identifying the answer.

The sentences were tokenized into their individual terms, and later the embeddings of each term were collated into one list of embeddings for the entire document.

We chose to take the approach of labeling all individual words within the document as being within the answer or not, and using that as the target of our model. We trained our model to predict which span of words in a document were the answer, rather than modeling answer classification sentence by sentence.

We tested two possible systems of labeling:

* An inside-outside (IO) labeling system in which answer words were labeled ‘IOA’ and non-answers labeled ‘OOA’
* A before-after-answer system where words before the answer sentence were labeled as ‘BA’, all words within an answer were labeled ‘IA’ and all words after an answer sentence were labeled ‘AA’.

Initially we began with an inside-outside-beginning-end (IOBE) classification, where the start and finish of the answer would be labeled as ‘BOA’ and ‘EOA’ respectively. However, we later decided that there would be insufficient support for the beginning and end tokens, and thus shifted to a simpler IO classification.

The second system trialed, classifying the words before and after the answer with two different labels was explored for a similar reason. Our first training run produced a model which predicted zero answer tokens, and while we later improved and refined our model training process, as discussed in subsequent sections, these results led us to explore alternative labeling systems with more even proportions of the classes. By splitting non-answer sentences into two groups, before and after answer, the proportions of labels in the data would not be as unbalanced as before.

1. Sequence QA model

We embedded each word using gensim’s pretrained glove-wiki-gigaword-100 model. This word embedder was chosen because it was a medium-sized model trained on 2014 Wikipedia and the Gigaword datasets. As this was partially trained on Wikipedia text, it was thought that its word embeddings may be more suitable than those of a model trained only on news or twitter corpora.

For feature extraction, we chose to perform part-of-speech tagging, term-frequency-inverse-document-frequency, named entity recognition, and implement a word match function.

Of these, the word match between question and answer seemed like it might be the most helpful. The question text was cleaned of common function words and simple punctuation using a list of ‘stop words’, before being passed to the Natural Language Toolkit WordNet lemmatizer to return a list of lemmas for content words within the sentence. The lemmatized words of the document were then compared to this list of question lemmas to identify which words matched with the lemma of a question word.

TF-IDF was performed on the document, using term frequency of a word within each sentence. Document frequency for the purposes of the calculation was number of sentences containing that word within the document.

Both the query and the document were tagged with POS and NER. POS tagging was performed using NLTK’s POS tagging model, with the output tags then converted to integer labels using a predefined index.

NER tagging was done using spaCy’s pretrained en\_core\_web\_sm model. Words were embedded with both the entity type and the entity IOB predicted by the model.

We exhaustively modeled all the different permutations of the input embedding variables in RNN models through a batch training process.

Best RNN IO models

While we tested a range of learning rates, from 0.001 to 0.1, the outcome of our batch testing was that the models with learning rate 0.1 had the better F1 performance for the test dataset.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Model** | **Attention** | **Doc word match** | **Q NER** | **Doc NER** | **Doc POS** | **Q POS** | **Doc TFIDF** | **Bidirectional** |
| **390** | Dot Prod | F | T | T | F | F | T | F |
| **1337** | Scaled Dot Prod | T | T | T | F | F | T | T |
| **1578** | Cos | F | T | F | F | FE | T | T |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Precision | Recall | F1 | Support |
| **390** | 0.5453 | 0.1102 | 0.1833 | 41,536 |
| **1337** | 0.4941 | 0.1122 | 0.1828 | 36,959 |
| **1578** | 0.6238 | 0.1063 | 0.1816 | 49,259 |

The optimal model found through our batch training did not include word matching in the input embedding. Examining the optimal model against the same model with word match added to the input embedding found a drop in F1 from 0.183 to 0.175. Looking at the models performances for the training data, it would appear that the model with word match fit slightly better to the training data than the model without word matching. It could thus be hypothesised that the decrease in test performance may be due to overfitting of the model to the training data.

Named entity recognition showed a clear improvement to the F1 of the model. It appears that removing NER from the question or document embedding was associated with an increase in precision but decrease in recall. But this relationship may be particular to this subset of features, and not carry over to other potential models.

Conversely, our optimal model did not include POS tagging of either the question or the document, and addition of POS tagging into that model showed a marked decrease in F1. It appeared that POS and NER may be performing a similar role, as removing NER from the document or question embedding would increase the performance of models using POS tags.

The following table was assembled using the best RNN IO mode that we found (bottom left) and the F1 scores for all models differing only in POS and NER embedding.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **F1 scores** | **No POS** | **Doc POS** | **Q POS** | **Doc and Q POS** |
| **No NER** | 0.1359 | 0.1365 | 0.1362 | 0.1353 |
| **Doc NER** | 0.1524 | 0.1623 | 0.1575 | 0.1485 |
| **Q NER** | 0.1567 | 0.1689 | 0.1345 | 0.1456 |
| **Doc and Q NER** | 0.1833 | 0.1630 | 0.1543 | 0.1393 |

In contrast, the best RNN BIA models were found to have utilized a different set feature embeddings than the RNN IO model. These best-performing RNN BIA models were as follows. Again, while we ran the models at 0.001, 0.01, and 0.1, all the best models had a learning rate of 0.1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **BIA model** | **Doc word match** | **Attention** | **Q NER** | **Doc NER** | **Doc POS** | **Q POS** | **Doc TFIDF** | **Bidirectional** |
| **591** | T | Cos | T | F | F | T | F | T |
| **475** | F | Dot Prod | F | T | F | F | T | T |
| **510** | T | Cos | T | F | F | F | T | F |
| **430** | T | Dot Prod | F | T | F | F | T | F |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Precision** | **Recall** | **F1** | **Support** |
| **Model 591** | 0.4690 | 0.1231 | 0.1950 | 31,966 |
| **Model 475** | 0.4495 | 0.1228 | 0.1929 | 30,727 |
| **Model 510** | 0.5941 | 0.1131 | 0.1900 | 44,076 |
| **Model 430** | 0.5513 | 0.1132 | 0.1878 | 40,871 |

While we were able to produce a better RNN model with this before-inside-after system, than with the IO labeling, the eventual GRU IO model we developed performed better than any of the GRU BIA models we were able to test.

We thus preceded with this RNN model embedding to test with models of different hidden layer and attention layers.

INSERT DETAILS OF THE MODEL ARCHITECTURE, ATTENTION ETC

1. Model Testing

Performance of the models were assessed by comparing the predicted classifications of each word in the document to the ground truth.

* 1. Input Embedding Ablation Study

Put your content here

* 1. Attention Ablation Study

Put your content here

* 1. Hyper Parameter Testing

Put your content here

For citations of references, we prefer the use of square brackets and consecutive numbers. Citations using labels or the author/year convention are also acceptable. The following bibliography provides a sample reference list with entries for journal articles [1], an LNCS chapter [2], a book [3], proceedings without editors [4], as well as a URL [5].

References

1. Yang, Y., Yih, W., Meek, C.: WikiQA: A Challenge Dataset for Open-Domain Question Answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 2013–2018. Association for Computational Linguistics, Lisbon (2015).
2. EXAMPLE CITATIONS FOR DELETION BELOW
3. Author, F.: Article title. Journal 2(5), 99–110 (2016).
4. Author, F., Author, S.: Title of a proceedings paper. In: Editor, F., Editor, S. (eds.) CONFERENCE 2016, LNCS, vol. 9999, pp. 1–13. Springer, Heidelberg (2016).
5. Author, F., Author, S., Author, T.: Book title. 2nd edn. Publisher, Location (1999).
6. Author, F.: Contribution title. In: 9th International Proceedings on Proceedings, pp. 1–2. Publisher, Location (2010).
7. LNCS Homepage, <http://www.springer.com/lncs>, last accessed 2016/11/21.