IIT-KGP at MEDIQA 2019: Recognizing Question Entailment using Sci-BERT stacked with a Gradient Boosting Classifier

Prakhar Sharma, Sumegh Roychowdhury
Indian Institute of Technology Kharagpur,
India
{prakharsharma, sumegh01}@iitkgp.ac.in

Abstract

The number of people turning to the Internet to search for a diverse range of health-related subjects continues to grow and with this multitude of information available, duplicate questions become more frequent and finding the most appropriate answers becomes problematic. This issue is important for question-answering platforms as it complicates the retrieval of all information relevant to the same topic, particularly when questions similar in essence are expressed differently, and answering a given medical question by retrieving similar questions that are already answered by human experts seems to be a promising solution. In this paper we present our novel approach to detect question entailment by determining the type of question asked rather than focusing on the type of the ailment given. This unique methodology makes the approach robust towards examples which have different ailment names but are synonyms of each other. Also it enables us to check entailment at a much more fine-grained level.

1 Introduction

Seeking health-related information is one of the top activities of todays online users via both personal computers and mobile devices. In all, 80 percent of Internet users, or about 93 million Americans, have searched for a health-related topic online, according to a study released on 16th July 2018 by the Pew Internet & American Life Project (Weaver, 2016) .Thats up from 62 percent of Internet users who said they went online to research health topics in 2001, the Washington research firm found. China (Guo et al., 2018) also has 194.76 million Internet health users in 2016, increased 28.0% compared with that in 2015. Despite the widespread need, the search engines often fail in returning relevant and trustworthy health information (Natalia et al., 2012)(Arabella et al., 2017). In this paper we try to bridge this gap by predicting entailment between questions. We particularly tackle this problem by checking entailment of a given consumer health question (CHQ) with most similar Frequently Asked Question (FAQ). Given two general English sentences this Question Entailment system can conclude whether answer of one question implies the other question's answer.

Q1: ”Can you mail me patient information about Glaucoma, I was recently diagnosed and want to learn all I can about the disease.”

Q2: ”What is glaucoma?”

In the above two questions the answer of Q1 implies the answer of Q2. (Entailment)

Detecting Question Entailment is a challenging task as it involves an amalgamation of tasks like Question Answering and Textual Entailment (Abacha and Demner-Fushman, 2019). Question answering is used to generate answers for both the questions and then checking textual entailment between the answers to give predictions possibly integrating Named Entity Recognition(NER) to our advantage. In this paper, we experiment on the MEDIQA 2019 task (Ben Abacha et al., 2019) by presenting an all-together different approach **QSpider** which overcomes these challenges by detecting question types instead of treating it like a pure Textual entailment or Question answering task.

Attempts have been made to tackle this problem, the most notable one being (Abacha and Demner-Fushman, 2016) which is the baseline for this task. The Baseline method uses supervised methods like SVM, Logistic Regression, Naive Bayes and used manual feature engineering but it fails to explore over the semantic space of the sentence. In this paper, we propose our model **QSpider** to tackle this problem.

QSpider is a staged system consisting of state-

Proceedings of the BioNLP 2019 workshop, pages 471–477
Florence, Italy, August 1, 2019. ©2019 Association for Computational Linguistics

of-the-art model **Sci-BERT** used as a multi-class classifier aimed at capturing both question types and semantic relations stacked with a **Gradient Boosting Classifier** which checks for entailment. QSpider achieves an accuracy score of **68.4%** which outperforms the baseline model (54.1%) by an accuracy score of 14.3%.

2 Related Work

2.1 Quora Question Pairs

Quora Question Pairs[1] is a binary classification task where the goal is to determine if two questions asked on Quora are semantically equivalent (Chen et al., 2018). Several works are done on this task with best performing ones being (MT-DNN (Liu et al., 2019), DIIN (Gong et al., 2018)). With MT-DNN's model incorporating a pre-trained bidirectional transformer language model similar to BERT (Devlin et al., 2018) while the fine-tuning part is leveraging multi-task learning. The DIIN model uses encoders to encode both the sentences and uses an interaction layer on top of it which is fed into a feature extraction layer. Finally the output layer decodes the acquired features to give predictions.

2.2 Recognizing Question Entailment

While textual entailment in open-domain has been extensively addressed in the literature, RQE has been less addressed for more restricted and specialized fields such as the medical domain. In *Recognizing Question Entailment for Medical Question Answering* (Abacha and Demner-Fushman, 2016) lexical features like Word Overlap and Bigram Similarity measures are used. It also tried to account for semantic features by using Negation Scope for Q1 and Q2, recognizing medical entities of 3 type: Problem, Treatment and Test. A different approach of using entailment in the QA problem is done in both the Pascal-RTE Challenge (Dagan et al., 2007), and in the CLEFAVE task (Kouylekov et al., 2006), by considering a question Q turned into an affirmative sentence as the hypothesis, and a text passage containing a candidate answer A as the text (i.e.systems have to decide whether A supports, or entails, Q).

3 Task Description & Dataset

The objective of this task is to identify entailment between two questions in the context of Question Answering. We use the following definition of question entailment: *Question A entails a Question B if every answer to B is also a complete or partial answer to A*. So, basically we need to predict, given two questions, if they entail each other or not.

The training corpus of MEDIQA 2019 RQE Shared Task (Ben Abacha et al., 2019) consists of 8,588 training pairs, containing 54.2% positive pairs. The remaining pairs (3,933) are negative examples collected by associating a random short form of NLM dataset question (JW et al., 2000) having at least one common keyword and at least one different keyword for each original question. The validation test corpus contains 302 pairs of questions consisting of 173 negative pairs and 129 positive pairs. Also the hidden test set had in total 230 pairs of questions of which 115 (50%) were true pairs and rest (115) false pairs. The question pairs in validation and hidden test set had its first question a Consumer asked Health Question (CHQ) and second question a Frequently Asked Question (FAQ). Upon doing an elementary analysis of the task dataset, we observe there are examples in validation and test set where medical entities are not in same form (either synonyms or abbreviation) in both questions but they still entail each other and vice versa.

Validation Set	**Positive**	**Negative**
Same Medical Entity	112	54
Different Medical Entity	17	119

Test Set	**Positive**	**Negative**
Same Medical Entity	87	101
Different Medical Entity	28	14

Table 1: Dataset Statistics : Positive means Entailment & Negative means Not Entailment.

We **additionally** used an annotated corpus of consumer health questions (Roberts et al., 2014) to build our question type prediction classifier. The corpus consists of 1,467 consumer-generated requests for disease information, containing a total of 2,937 questions. The dataset has these requests classified into 13 question types or classes namely: *Anatomy, Cause, Complication, Diagnosis, Information, Management, Manifestation,*

[1] https://www.kaggle.com/c/quora-question-pairs

Other effects, PersonOrg, Prognosis, Susceptibility, Other, Not Disease.

4 Models

In this section we will discuss about the various approaches we have used for building our Question Entailment detection model.

- **Dependency Tree-LSTM** (Tai et al., 2015): A generalization of LSTM (Long Short-Term Memory) to tree-structured network topologies. The model was aimed to capture the syntactic relations between two questions.
- **BERT$_{\textbf{Large, uncased}}$** (Devlin et al., 2018): BERT which stands for Bidirectional Encoder Representations from Transformers is designed to train deep bidirectional representations by jointly conditioning on both left and right context in all layers. Language models have demonstrated that rich, unsupervised pre-training is an integral part of many language understanding systems. Hence, we try fine-tuning BERT to obtain better results on this task.
- **Bio-BERT** (Lee et al., 2019): Domain specific language representation model based on BERT and pre-trained on large-scale biomedical corpora.
- **Sci-BERT + Hinge loss** (Beltagy et al., 2019) : A pre-trained contextualized embedding model based on BERT to address the lack of high-quality, large-scale labeled scientific data, fine-tuned with a Hinge loss function. This outperformed all other systems during the validation phase.

Now we describe all our approaches in-detail.

4.1 Dependency Tree-LSTM

We refer to a Child-Sum Tree-LSTM(Tai et al., 2015) applied to a dependency tree as a Dependency Tree-LSTM. We produced dependency parses[2] of the questions in the dataset for our Dependency Tree-LSTM model. Each Tree-LSTM unit (indexed by j) contains input and output gates i_j and o_j, a memory cell c_j and hidden state h_j. The difference between the standard LSTM unit and Tree-LSTM units is that gating vectors and memory cell updates are dependent on the states of possibly many child units. Additionally, instead of a single forget gate, the Tree-LSTM unit contains one forget gate f_{jk} for each child k. This allows the Tree-LSTM unit to selectively incorporate information from each child. Each Tree-LSTM unit takes an input vector x_j. We took, each x_j as a vector representation of a word in a sentence. The input word at each node depends on the tree structure used for the network.

We first produce sentence representations h_L and h_R for **question1** and **question2** respectively in the pair using a Tree-LSTM model over question's parse tree. Given these sentence representations, we calculate the entailment probability $\hat{p}_\theta$ using a neural network that considers both the distance and angle between the pair (h_L, h_R):

$$\mathrm{h}_\times = h_L \odot h_R,$$

$$\mathrm{h}_+ = |h_L - h_R|,$$

$$\mathrm{h}_s = \sigma\left(W^{(\times)}h_\times + W^{(+)}h_+ + b^{(h)}\right),$$

$$\hat{p}_\theta = \mathrm{softmax}\left(W^{(p)}h_s + b^{(p)}\right),$$

We want $\hat{p}_\theta$ given model parameters θ to be close to the p. Here y denotes whether it is an entailment. Hence we decide the cost function as the regularized KL-divergence between p and $\hat{p}_\theta$:

$$\mathrm{p}_i = || \; i - y \; | - 1 \; | \qquad i = \{0, 1\}$$

$$\mathrm{J}(\theta) = \tfrac{1}{m}\textstyle\sum_{k=1}^{m} \mathrm{KL}\left(p^{(k)} \,\middle\|\, \hat{p}_\theta^{(k)}\right) + \tfrac{\lambda}{2}\|\theta\|_2^2,$$

where *m* is the number of training pairs and the superscript *k* indicates the k-th sentence pair.

4.2 BERT

We chose BERT$_{\text{Large, uncased}}$ as our underlying BERT model. It consists of 24-layers, 1024-hidden, 16-heads, and 340M parameters. It was trained on the BookCorpus (800M words) and the English Wikipedia (2,500M words). The two input sentences in form of **question1** and **question2** were first tokenized with the BERT basic tokenizer to perform punctuation splitting, lower casing and invalid characters removal. The maximum sequence length was defined as 128, with shorter sequences padded and longer sequences truncated to this length.

[2] Dependency parses produced by the Stanford Neural Network Dependency Parser (Chen and Manning, 2014)

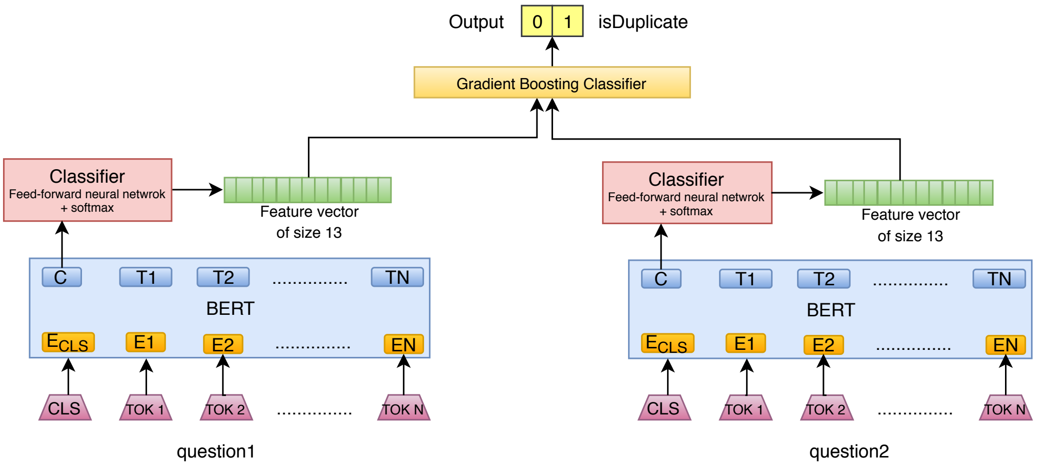

Figure 1: Model - QSpider

We used the PyTorch implementation from *pytorch-pretrained-bert*[3] which had the BERT tokenizer, positional embeddings, and pre-trained BERT model. Following the recommendation for fine-tuning in the original BERT approach (Devlin et al., 2018), we trained our classifier with a batch size of 32 for 5 epochs. The dropout probability was set to 0.1 for all layers, and Adam optimizer was used with a learning rate of 2e-5 with Binary Cross Entropy Loss as the loss function defined below:

$$\text{L}_{entropy}(x, class) = -\log\left(\frac{\exp(x[class])}{\sum_j \exp(x[j])}\right)$$

4.3 Sci-BERT + Hinge loss

We then tried using domain specific variants of BERT such as Bio-BERT (Lee et al., 2019) and Sci-BERT (Beltagy et al., 2019). Bio-BERT was pre-trained on biomedical domain corpora (e.g., PubMed abstracts, PMC full-text articles), whereas Sci-BERT consists of a custom-made vocabulary (Sci-Vocab) which consists of frequently observed words and subwords in scientific text which may differ from those occurring in general domain text. Sci-BERT outperformed Bio-BERT in this task. Since for binary classification tasks, both Hinge Loss and Cross-Entropy Loss are widely used, we tried incorporating both of these losses in our model. In this task, Hinge Loss did give a better accuracy as reported below. Hence, we focused on finetuning Sci-BERT by changing the loss function from Binary Cross Entropy to Hinge Loss(used in SVMs) which resulted in an increase of accuracy approximately by 2% on the validation set.

$$\text{L}_{hinge}(f) = \textstyle\sum_{j \neq y_i} max(0, s_j - s_{y_i} + 1)$$

We discuss further about this in later sections.

5 QSpider - System Description

QSpider is a staged system consisting of Sci-BERT (Beltagy et al., 2019) stacked with a Gradient Boosting Classifier which performed the best in the hidden test set among all the models described above. This model aims at capturing question types and use them as features to detect question entailment. We trained a multi-label classifier (as a question can fall in more than one class) on a annotated corpus (13 question types as mentioned above in *Section 3*) of consumer health questions (Roberts et al., 2014). For example:

Q: ”Can you mail me patient information about Glaucoma, I was recently diagnosed and want to learn all I can about the disease.” .

Qtype: ”Information” .

Since the available annotated dataset was not sufficient to build our question type classifier model hence we used pre-trained language model to efficiently learn from this small dataset. We used Sci-BERT as language model here as it can easily detect the semantic feature of question. After training on this dataset we predicted on our original Train, Validation and Test dataset.

We used Scibert-scivocab-uncased as the vo-

[3] `https://github.com/huggingface/pytorch-pretrained-BERT`

cabulary for our model. A vector of 1's and 0's of length 13 (number of question types) was obtained for each question in our Train, Validation and Test set. We horizontally stacked these vectors for **question1** and **question2** and used them as feature vector of shape 26 for our next model. Next we use these feature to train our **Gradient Boosting Classifier**[4], which predicts whether the two questions are an entailment or not. We further fine-tuned our Gradient Boosting Classifier by keeping the number of estimators as 5000, to obtain the optimal performance on our hidden Test set without overfitting.

6 Results

This section discusses regarding the results of various approaches we applied in this task. Since the training data-set (Ben Abacha et al., 2019) had less training examples, the systems were made to learn from the training data and tested on the validation data for validation results while for test results the systems learned from the training + validation data and tested on training data. *Table 2* represents the accuracy of the systems described on the validation and test data.

Taking *BERT*$_{large}$ as our baseline, it gives an accuracy of 76.2% outperforming Tree-LSTM (64%) and QSpider (62.0%) on validation set. The more domain-specific models like Bio-BERT (77.6%) and *Sci-BERT + Hinge Loss (80.5%)* gave a significant boost. Also, Sci-BERT + Hinge Loss was the **best performing system** among all participants during Validation phase. For the test set BERT$_{\text{large}}$ gives an accuracy of *48.1%* and similar models like Bio-BERT and Sci-BERT + Hinge Loss gives an accuracy of 49.6% and 51.3% respectively. Here the more syntatic models like Tree-LSTM (60.2%) perform much better. Our model *Qspider (68.4%)* performs the best here and **3rd** overall among all participating systems.

7 Error Analysis

The training examples were much easy to check for entailment, with most of the positive pairs having common sub-strings or having similar syntactic structure. As discussed earlier in *Section 3*, in the validation set, out of 302 examples, 112 examples had same same medical entities which also entail each other and 119 examples with different medical entities which do not entail each other (refer *Table 2*) because of which attention models like BERT gained a huge success by focusing more on entity name. It is also evident (refer *Table 3*) that these models fail in those cases where there is same medical entity on both sides but the pair is not an entailment.

Model	**Valid**	**Test**
Tree-LSTM	64.0	60.2
BERT$_{\text{large, uncased}}$	76.2	48.1
Bio-BERT	77.6	49.6
Sci-BERT + Hinge Loss	**80.5**	51.3
QSpider	62.0	**68.4**

Table 2: Accuracy results for various models.

Validation Set	**Correct**	**Wrong**
Same Medical Entity (Positive)	112	0
Same Medical Entity (Negative)	1	53
Different Medical Entity (Positive)	13	4
Different Medical Entity (Negative)	117	2
Test Set	**Correct**	**Wrong**
Same Medical Entity (Positive)	87	0
Same Medical Entity (Negative)	1	100
Different Medical Entity (Positive)	24	4
Different Medical Entity (Negative)	6	8

Table 3: Number of Correct and Wrong predictions made by Sci-BERT on the task dataset. Positive means Entailment & Negative means Not Entailment.

The Hidden Test set had more than 80 % pairs (refer *Table 2*) where there are same medical entity in both questions but still more than 50% pairs among these does not entail each other. Remaining examples are even more complicated like pairs having medical entity names as synonyms/abbreviated forms of each other. **This caused a huge drop in accuracy of attention based models like BERT**. Here is where QSpider comes to the rescue, by not only focusing on syn-

[4] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.GradientBoostingClassifier.html

tactic but also on semantic to capture the type of question asked and also not giving high attention to entity name.

QSpider on the other side didn't perform equally well in the validation set since there are considerable number of examples having different medical entities in **question1** and **question2**. We didn't give any attention to entity name while designing QSpider keeping in mind the Test set. This is the reason QSpider doesn't perform well on the Validation set but gives good results on the Test set.

8 Conclusion and Future Work

In this paper we discussed regarding various deep learning approaches and our final model **QSpider**. It is evident from the results that even with very small sized data type we were able to generate satisfactory predictions for question type. There is a scope of improvement with the increase in the question type data. We can see that question type plays an important role in capturing question entailment but if the questions has same type but different medical entity name then our system might mis-classify. Since, our Test dataset didn't have such examples with different medical entities, hence we didn't integrate this with **QSpider** then.

We plan to integrate our model with detection of medical entity names of the questions and append them to our existing feature vector to capture difficult examples. Currently, we are using question types as discrete and independent classes which we pass onto the Gradient Boosting Classifier. But in reality, any question asked cannot be always classified into a particular question type. It always consists of a blend of various types of question. So we plan upon using the GloVe embeddings of the question classes (as mentioned in above sections) as extended features to be passed onto our classifier.

9 Acknowledgements

We would like to thank everyone in the organizing committee and the reviewers for reviewing our paper and providing their valuable feedback. We also acknowledge support from the **Computer Science and Engineering Department** of the **Indian Institute of Technology Kharagpur, India** for providing us with the computational resources required for running our models for this task. Lastly, we would also like to thank **T.Y.S.S. Santosh**[5] for proof-reading the paper and providing other valuable suggestions towards improving the paper.

References

Asma Ben Abacha and Dina Demner-Fushman. 2016. In recognizing ques- tion entailment for medical question answer- ing.

Asma Ben Abacha and Dina Demner-Fushman. 2019. A question-entailment approach to question answering.

Scantlebury Arabella, Booth A, and Hanley B. 2017. Experiences, practices and barriers to accessing health information: A qualitative study.

Iz Beltagy, Arman Cohan, and Kyle Lo. 2019. Scibert: Pretrained contextualized embeddings for scientific text.

Asma Ben Abacha, Chaitanya Shivade, and Dima Demner-Fushman. 2019. Overview of the mediqa 2019 shared task on textual inference, question entailment and question answering, acl-bionlp 2019. In *Proceedings of the BioNLP 2019 workshop, Florence, Italy, August 1, 2019*. Association for Computational Linguistics.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks.

Z. Chen, H. Zhang, X. Zhang, and L. Zhao. 2018. Quora question pairs.

Ido Dagan, Bill Dolan, Danilo Giampiccolo, and Bernardo Magnini. 2007. The third pascal recognising textual entailment challenge.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Yichen Gong, Heng Luo, and Jian Zhang. 2018. Natural languagev inference over interaction space.

Haihong Guo, Xu Na, and Jiao Li. 2018. Qcorp: an annotated classification corpus of chinese health questions.

Ely JW, Osheroff JA, Gorman PN, Ebell MH, Chambliss ML, and Pifer EA. 2000. A taxonomy of generic clinical questions: classification study. *British Medical Journal*, 321:429–432.

Milen Kouylekov, Matteo Negri, Bernardo Magnini, and Bonaventura Coppola. 2006. Towards entailment-based question answering: Itc-irst at clef 2006.

[5] https://scholar.google.ca/citations?user=aYytWsAAAAAJ&hl=en

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2019. Biobert: a pre-trained biomedical languagerepresentation model for biomedical text mining.

Xiadong Liu, Pengcheng He, Weizhu Chen, and Jianfeng Gao. 2019. Multi-task deep neural networks for natural language understanding.

Pletneva Natalia, Vargas A, Kalogianni K, and Boyer C. 2012. Online health information search: what struggles and empowers the users?

Kirk Roberts, Kate Masterton, Marcelo Fiszman, Halil Kilicoglu, and Dina Demner-Fushman. 2014. Annotating question types for consumer health questions.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks.

Jane Weaver. 2016. More people search for health online.

A Supplemental Material

This is the link to our classifier code for **QSpider** which can be used to reproduce the results claimed in the Results section above for **QSpider** - https://github.com/Team-IIT-KGP/Qspider. The README section is updated for instructions to run the code.