A Sentiment Based Non-Factoid Question-Answering Framework

Conference Paper · October 2019

DOI: 10.1109/SMC.2019.8913898

2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)
Bari, Italy. October 6-9, 2019

A SENTIMENT BASED NON-FACTOID QUESTION-ANSWERING FRAMEWORK

Qiaofei Ye, Kanishka Misra, Hemanth Devarapalli, and Julia Taylor Rayz
Purdue University, West Lafayette - IN 47906

***Abstract*— With the rapid advances in Artificial Intelligence, a question of emotional intelligence of a system may become as important as its accuracy. This paper investigates whether emotions should be considered for non-factoid "how" Question-Answering systems with the eventual goal of enabling the system to retrieve answers in a more emotionally intelligent way. This study proposes an architecture that adds extended representation of sentiment information to questions and answers, and reports on to what extent a prediction of the best answer be improved by the proposed architecture.**

I. INTRODUCTION

With the rapid growth and maturity of Question-Answering (QA) domain, non-factoid Question-Answering tasks are getting more and more attention. However, most of the existing Question-Answering systems are either fact-based or highly keyword related and hard-coded. Fact-based QA means there is only one correct answer which is based on publicly-known fact for the question. Highly keyword related and hard-coded QA means the system returns the pre-defined answer based on the keyword inserted.

In real-world applications, people often encounter situations where a posed question does not get an expected or a desirable answer. One of the reasons behind is that in person-to-person communication, open-ended questions, such as why-question or how-question, often follow the pattern that "if something undesirable happens, the reason is also often something undesirable, and if something desirable happens, the reason is also often something desirable" [1], thus taking sentiment into account. However, when a machine selects an answer to a question that a person asked, the sentiment is usually disregarded. This may lead to a user that regards an answer as inappropriate even if it is technically correct.

For the reason mentioned above, if QA is to become more personable, the sentiment of the question and answer should be taken into account. Mishra [2] stated that WHY questions asked in Opinion Question-Answering systems (OQAS) are rated higher when answers incorporate reasons and explanations for the questioners' sentiment expressed in the questions [2]. For this reason, when the Question-Answering System is designed, not only the quality of the fact or information contained in the answer should be considered, but also how the sentiment is addressed. Thus, when having several candidate answers in the answer pool that belong to the same question, knowing how to choose the answer that both indicates correct information and contains proper sentiment may be of great importance in non-factoid Question-Answering tasks.

The goal of this research is to improve the Question-Answering framework to retrieve the best answer in a more emotionally intelligent way. More specifically, in non-factoid Question-Answering domain, this study investigates whether adding sentiment into non-factoid Question-Answering can help improve the performance of retrieving the best answer.

The remainder of this paper is organized as follows: Section II summarizes related research in non-factoid QA and sentiment framework. Section III describes the proposed method, and is followed by Section IV that reports the results.

II. BACKGROUND

A. *Non-Factoid Question-Answering systems with Answer Ranking*

Most of the state-of-the-art Question-Answering (QA) systems serve for answering fact-based questions such as "When was Steve Jobs born?" and "Who is the current president of the US?" In addition to facts, in various scenarios, people sometimes like to know about others' opinions, ideas, and feelings of some specific topics. This kind of non-fact-based QA system is called Non-Factoid QA.

Previous work on answer selection for non-factoid QA usually adopted approaches like feature engineering [3], linguistic tools [4], or some other external resources [5]. The answer selection problem could be transformed into a syntactical task [5], performing matching between the question-answer pairs parse trees. Eskandari [6] adopted information generated from sentiment analysis (SA), spell checking, and also social network behaviors, like votes with user information, to predict the best answers. Methods that do not rely on external resources in non-factoid QA have also been tried. For example, Feng [7] proposed a deep learning framework with CNN structures, the best answer is selected based on the similarity of generated vector representation for both question and answer, with a top-1 accuracy of 58.2%. Based on Feng's [7] work, Tan [8] developed an approach with a neural network based on bidirectional LSTM and CNN for non-factoid answer selection. The new network results in 3.7% higher accuracy over the selected baseline [7].

B. *Deficiency in Non-Factoid Question-Answering using sentiment*

Sentiment analysis and classification is a problem that has been applied to many domains, but rarely to non-factoid Question-Answering. Some research, however, found that the addition of sentiment could be useful.

978-1-7281-4569-3/19/$31.00 ©2019 IEEE

Oh [1] first introduced sentiment analysis to non-factoid Question-Answering by using sentiment analysis and word classes for ranking answers to WHY-questions in Japanese. Oh's work is based on combination of sentiment polarity and the contents of sentiment expressions associated with the polarity in questions and their answer candidates. It gains 15.2% improvement in precision at the top-1 answer over the baseline state-of-art QA system at that time [1]. This research indicates that in the domain of open-ended questions, using sentiment and other Natural Language Processing features can achieve a likely gain in QA systems compared to simple fact-based Question-Answering without using sentiment.

Ku [9] presents an Opinion Question-Answering framework that aims at question analysis and retrieving answers from passage. They conclude that the best answers sometimes have sentiment correlation with the question [9]. For opinion answer retrieval tasks, they were concerned not only the relevance but also the sentiment. According to [9], considering both opinion and action words are better than opinion words only. The paper focuses on sentiment information on word-level.

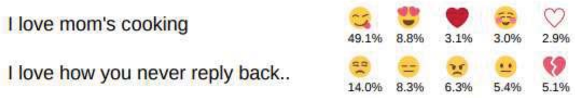
Eskandari [6] proposed a design for predicting the best answers in Community Question-Answering systems based on sentiment. In this experiment, the Sentiment Analysis (SA) and subjectivity/objectivity identification are used to classify a given text positive, negative or neutral and classes objective or subjective. This work considers comments as one of the inputs. By finding the best combination of different features, the new model outperforms the baseline by 2% to 6%.

C. Existing Sentiment and Subjectivity Analysis framework

Sentiment could be classified among various dimensions, from binary (positive/negative) and ternary (positive/neutral/negative) to any number N that would represent the range of scale or categories [10]. A single system could be built to perform the task, or integrated sentiment analysis frameworks have been proposed. The Stanford CoreNLP toolkit [11], is an extensible framework that provides core natural language analysis. This toolkit is widely used in the research NLP community and also among commercial and government users of open source NLP technology [11]. For sentiment analysis, it can categorize the sentence into 'positive,' 'neutral,' and 'negative' categories.

Some sentiment analysis frameworks do not provide sentiment classification. Instead, they provide sentiment information in an extended scale [12]. Such approach allows avoiding the limit "of the scarcity of manually annotated data" [12] by extending the distant supervision to a more diverse set of noisy labels of 64 dimensions. The models can learn richer representations compared to information containing only positive/negative/neutral categories. Deepmoji [12] trained the model for emoji prediction on a dataset of 1246 million tweets containing one of 64 popular emojis. The model obtained 82.4% agreement of the tweet's polarity in emotion detection. For each input sentence, there is a 64-dimension output vector, representing the confidence of each emoji. An example of the corresponding input and output of this model is provided in Figure 1, only the top-5 emojis with the highest confidence scores are shown in this figure.

Figure 1. Examples of Deepmoji results [12]



Subjectivity detection and computation has also emerged as a research topic. For example, Textblob [13] text analysis framework is used to generate subjective scores of an input sentence [6], with a subjectivity scale from 0 to 1.

III. NON-FACTOID QA WITH SENTIMENT MODEL

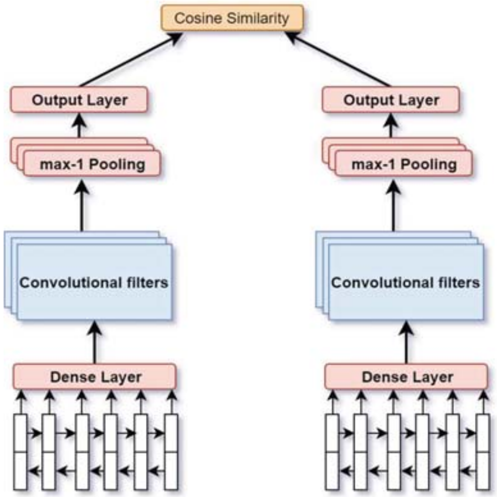
In this section, the proposed method is presented. The methodology of this study includes Sentiment Information Computation and new neural network structure construction with sentiment. The proposed workflow is presented.

A. Neural network structure construction with sentiment

For this study, [8] is selected as the baseline model as it "substantially outperforms several strong baselines" [8]. At the same time, framework proposed in [8] does "not rely on external resources", thus "can be easily adapted to different languages or domains" [8].

The baseline model [8] is a bidirectional LSTM network with a CNN on top of it. The best answer is selected based on cosine similarity of question and answer pairs. The answer with the highest similarity is selected as the best answer. The structure of this baseline model is visualized in Figure 2.

Figure 2. Network structure of baseline model



The initial word embeddings that serve as input to the LSTM network were trained by word2vec [14]. Word embeddings are also parameters in the training process and were optimized according to input data during training.

The LSTM is applied separately to Word Embeddings of question and answers to get a more precise representation of time sequences of sentences, creating hidden vectors for the question and answers. The hidden states of the network serve as input to a CNN architecture, which provides ranking to available answers. At the same time, it also provides a more composite representation of questions and answers [8]. Compared with evenly considering the lexical information of each token, this architecture emphasizes certain parts of the answer, in order to differentiate the incorrect answers from the ground truth answers.

The LSTM hidden state was set to size 200 for one direction. Four kinds of CNN filters were applied upon the hidden state with window sizes of 1, 2, 3, 5, number of filters for each kind is set to 500. Max-1 pooling layers are applied after the CNN structure. The final output is a 2000-dimensional output vector for comparing the similarity between questions and answers.

In the original model, after generating the output vector for comparing similarity, a pairwise ranking method was adopted to define the objective function. For question and each of its candidate answers, a question-answer pair is constructed. The similarity was computed for each input pair. The answer with the highest similarity score is selected as the best answer.

The loss is computed by the distance for each question-answer pair as shown below:

$$L = \max\{0, \lambda - sim(q, a_+) + sim(q, a_-)\} \qquad (1)$$

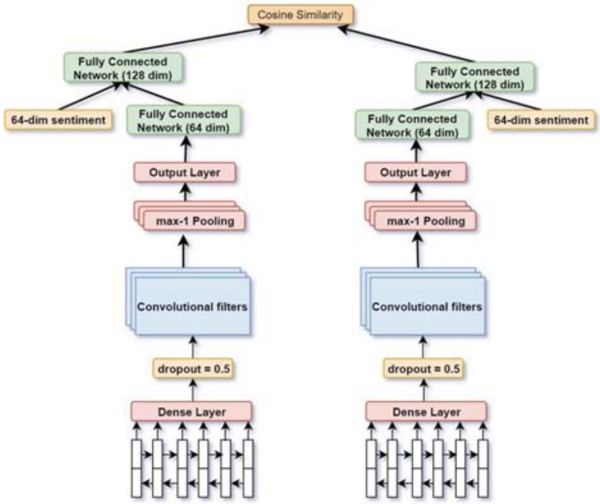
where a+ is the ground truth, which can be thought of the answer with the annotation of "best answer" from the dataset, a− is an incorrect answer from other candidate answers belonging to the same question. For this study, the incorrect answer is selected from the answer space under the same question except for the answer marked as 'best answer'. The baseline model only uses the answer pair with the highest L to update the weight.

For the new network architecture, the baseline network is augmented with sentiment information when performing the best-answer selection. The baseline model aims at extracting information based on text, and it is desirable to keep this architecture in the new model. The sentiment information is computed in sentence-level, thus it is added as extra information after the text information is computed and adjusted. The only adjustment to the text information processing structure is to add a dropout layer (set to 0.5) after the LSTM hidden states to prevent overfitting. The input to the pre-trained Deepmoji model [12] is the questions and their corresponding candidate answers. The 64-dimension sentiment information is generated for each question and answers in this dataset, stored to be used as the input to the new network.

In order to assign equal weights to text information contained in the question and answers, and the corresponding sentiment information contained within them, the output of CNN is passed into a Fully Connected Network, which is constructed of a Dense Layer, one activation layer with ReLU activation function, and another Dense Layer. The ReLU activation function is selected because ReLU offers faster-converging speed, and also is more capable of reducing the possibility of vanishing gradients in the training process. The output of this Fully Connected Network (FCN) is a 64-dimension vector, which is the same dimension as the sentiment vector.

The 64-dimension vector of FCN is concatenated with the 64-dimension sentiment vector, resulting in a 128-dimension joint vector. This joint vector is passed to another Fully Connected Network (FCN), with the output being a vector of 128-dimension vector. The size 128-dimension is chosen with the concern of reducing dimension sometimes leads to information loss, and we experimented that the output dimension less than 128 lead to a worse performance. This Fully Connected Layer (FCN) can make the model learns the text feature and sentiment feature together, in the process of tuning the network weight. The new neural network architecture is shown in Figure 3. We use the same loss function as the baseline.

Figure 3. New Neural Network Architecture



B. Evaluation

Evaluation was performed by Precision of the top answer (P@1), and Mean Reciprocal Rank (MRR).

P@1 is the precision of the top answer, measuring how many questions have a correct top answer candidate. In this case, since we only have one predicted best answer, and only one ground-truth best answer, P@1 is also equal to accuracy in this case.

MRR matrix is used for evaluating any process that produces a list of possible responses to a sample of queries Q, ordered by the probability of correctness. In equation (2), i stands for the i-th query, rank means the position of the first relevant answer.

$$\text{MRR} = \frac{1}{|Q|}\sum_{i=1}^{|Q|}\frac{1}{rank_i} \qquad (2)$$

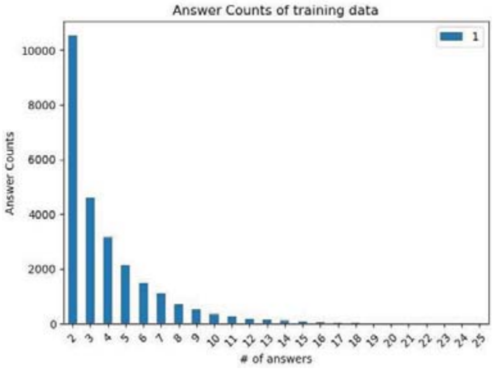
A paired T-Test with significance level of 0.05 is employed to calculate the statistical significance between the new neural network, and the baseline.

C. Dataset

The dataset used for this study is L5 - Yahoo! Answers Manner Questions, version 2.0 from Yahoo research. The data collected is a subset of the Yahoo! Answers corpus from 10/25/2007 that starts with the word "How". For example, all the questions start with "How", following any word from the list "to," "do," "did," "does," "can," "would," "could," and "should." In this dataset, only questions and answers that have at least four words, out of which at least one is a noun and at least one is a verb, were kept. Questions and answers of apparent low quality were removed. The resulting dataset contains 142,627 questions and their answers, annotated for best answer selection, and category and sub-category of questions.

Due to computational complexity and speed of the biLSTM model, only the first 30% of the data is used for this study, resulting in 29951 number of questions. For the selected data, the average number of answers per question is 4.13 excluding the question with only one answer (4111). The distribution of number of answers per question is shown in Figure 4 for the selected dataset. In Figure 4, questions with an answer count equal or lower than five are left out. The data are then split into training, validation, and testing set, using 70:15:15 split. The number of questions in training set is 20965, for validations set is 4492, for the testing set is 4493. The same dataset was used for baseline and new network comparison.

Figure 4. Answer count distribution of the selected dataset



D. Data Preprocessing and Assembly

Several annotated tags in the corpus are useful:

- "subject" means questions;
- "bestanswer" means the annotated best answer among all the candidate answers;
- "nbestanswers" tag has several sub-tags under it, the sub-tag is called "answer_item", which corresponds to one candidate answer item.

For traditional QA tasks, punctuations are removed since those studies aim at capturing the lexical information contained in sentences. In this study, the punctuations were kept due to the importance of punctuations in sentiment analysis. Non-ASCII characters were removed. Additionally, phone numbers, continuous white spaces, and URLs were also removed. Words directly connected with the previous word through a period in between had a space inserted after the period to separate them. For example, originally the tokenized result containing records like 'kitchen.I' were corrected to 'kitchen. I'. The hyphens existing on one end or both ends of a word were removed. For example, originally tokenization results containing words like: 'nonrainy-' or '-obey' had hyphens removed. At the same time, continuous hyphens with different length were also removed due to replication in tokenization.

The selected dataset is divided into training, validation, and testing set by a ratio of 0.7, 0.15, and 0.15. The dictionary of the entire data space was generated by Natural Language Toolkit (NLTK). Tokenization was performed on the training, validation, and testing set, using TweetTokenizer from NLTK. The final encoded data were generated based on the dictionary id. Questions and answers are padded into a length of 150, and if the length of the question/answer is longer than 150, only the first 150 words are selected. After this, the cleaned data is ready to be imported into the model. Things to be noticed is that the incorrect answers in the baseline model [8] are randomly selected from the answer pool of the entire dataset, which could include answers intended for different kinds of questions in different domains. Instead, in our experiment, we only consider incorrect answer under the same question, which is more reasonable for an answer ranking task.

The framework generates a 64-dimension confidence list corresponding to 64 different emojis for each question and answer. The full list of emojis can be found in the Deepmoji paper [12]. As the experiment iterated over the dataset, corresponding sentiment information aligning with the text of each question/answer were received.

E. Experiment Details

The baseline model is trained on 20965 questions and their corresponding answers (20965 best answers and 106733 incorrect answer). All experiments are processed on a GPU cluster. We use the P@1 measure to check the performance of the model on the validation set to locate the best epoch and best hyper-parameter settings.

The model is trained in a batch of 64, and the maximum length of questions and answers is 150 (only the first 150 words of each question and answer will be used for training), any tokens out of this range was discarded. The initial word embedding was trained using word2vec [14] with the word vector size set to 100. Word embeddings are also parameters in the training process and were optimized according to input data during training, with Rmsprop as the optimization strategy. The margin values are set to 0.2.

F. Performance Testing

The baseline model and the proposed architecture with the sentiment information is evaluated on the validation set with 4492 questions and test dataset with 4493 questions, the Precision@1 and the Mean reciprocal rank (MRR) scores are calculated and compared with the baseline performance.

IV. RESULTS AND DISCUSSION

A. Performance on validation and testing set

The results for the proposed new neural network architecture and the baseline are reported in Table 1 (statistically significant results are marked with *):

TABLE I. PERFORMANCE COMPARISON ON THE VALIDATION SET AND TESTING SET

Models	Performance Metric			
	Precision@1 on validation set	***MRR on validation set***	***Precision@ 1 on test set***	***MRR on test set***
Baseline	0.4484	0.6387	0.5593	0.7395
Model with sentiment	**0.5859***	**0.7504***	**0.5718***	0.7379

As can be seen, the new model outperforms the baseline. The Precision@1 performance gain reached 13.76% for the validation set, and 1.25% for the testing set. While the baseline model performance varies by 11.10% between validation set and testing set, the new model has a more consistent performance on the validation set and test set.

However, in order to understand the difference in performance between validation and testing, we performed further experiments.

B. Performance on sub-tests

A testing scheme was designed to test the performance difference between the neural network with sentiment and the baseline. This set of tests are focused on two variables: sentiment and subjectivity. The hypothesis is that since subjective questions request a person's personal opinion, sentiment could play a heavier role for these questions. The new test suite is divided into four tests listed below:

(1) Questions without sentiment versus questions with sentiment;

(2) Subjective questions versus non-subjective questions;

(3) Subjective questions with sentiment versus all subjective questions;

(4) Non-subjective questions with sentiment versus all non-subjective questions.

In order to perform this set of tests, the dataset should be further annotated to select the subset for question-answer pairs whose question contains sentiment or does not contain sentiment, and also the subset of question-answer pairs whose question belongs to subjective questions or non-subjective questions.

We assume that both positive or negative questions or answers contain sentiment, thus we differentiate these from neutral questions or answers. Furthermore, we only consider these three categories for our classification instead of 64-dimention.

For questions containing or not containing sentiment, the Stanford core NLP sentiment analysis framework [10] is employed to classify the input sentence into 'neutral', 'positive', or 'negative'. We combine 'positive' and 'negative' questions as the category 'questions with sentiment', and 'neutral' as ''questions without sentiment.

For subjective question and non-subjective question, the Textblob [13] text analysis framework is employed to generate the subjectivity score from 0 to 1. Subjectivity score 0 means non-subjective and 1 means subjective. A question with a subjectivity score higher than 0.6 is considered subjective question, at the same time, a question with a subjectivity score less than 0.2 is considered a non-subjective question for this study.

Because the ratio of the questions with sentiment is relatively small compared with the questions without sentiment in this dataset, the number of questions with sentiment is not enough for testing if the data is only retrieved from the testing set. The same circumstance applies to subjective questions. To address this concern, the data used for this set of tasks are retrieved from the rest 70% of the dataset, which is not originally used for training, validation, and testing set. The first 4000 questions for each category in sub-test (1) and (2) are selected and shuffled, ready to be used for the final testing. The questions in subjective classification with sentiment category are retrieved by performing the intersection of 4000 subjective questions and 4000 questions with sentiment, resulting in 511 questions. The questions in non-subjective set with sentiment are retrieved by performing the intersection of 4000 non-subjective questions and 4000 questions with sentiment, resulting in 436 questions. The evaluation follows the same metrics as for the overall architecture performance (Precision@1 and MRR). The comparison of results for four sub-tests can be found in Table II.

TABLE II. PERFORMANCE COMPARISON ON FOUR SUB-SET

Sub-Test	Performance Metric			
	Precision @1 of baseline	***MRR of baseline***	***Precision@1 of new model***	***MRR of new model***
Questions with sentiment	0.4616	0.6538	**0.4758**	**0.6643**
Quesitions without sentiemnt	0.4971	0.6881	**0.5236***	**0.7041***
Subjective Questions	0.4457	0.6418	**0.4581**	**0.6473**
Non-subjective Questions	0.5042	0.6939	**0.5319***	**0.7100***
Non-subjective Questions with Sentiment	0.4971	0.6881	**0.5586**	**0.7315***
Subjective Questions with Sentiment	0.4266	0.6229	**0.4289**	0.6217

For sub-test (1), the performance for questions without sentiment is better than questions with sentiment for both the baseline and the new model, also the improvement rate is higher for questions without sentiment. This may be due to the fact that the quantity ratio of questions with sentiment and questions without sentiment is 1:6.98 in the training set. Not enough training samples for the questions with sentiment is one possible reason why performance in this category is lower.

Second, the sentiment framework [11] we adopted for pruning the data into 'questions with sentiment' and 'questions without sentiment' is not the same as the sentiment framework [12] we adopted for generating sentiment information within the network for each question/answer. The difference in their design/judging criterion could lead to a different sentiment evaluation result. While both of questions with sentiment and without sentiment showed performance gain, we can conclude that, add the extended sentiment components into question answering can improve the performance, even for neutral questions.

For sub-test (2), the reason why performance in subjective questions category is lower could also belong to not having enough training samples for subjective questions. On the other hand, since the methodology we adopted is a similarity-based method, it may not work well on subjective questions. The non-subjective question would perform better since the answer could have a higher word overlap with the question, as it usually point out the answer in the same domain as the question proposed, while subjective questions could extend the domain to other related topics, thus have a lower word overlap.

For sub-test (3), while sub-test one and two indicate that subjective questions and questions with sentiment do not perform well on its own category, sub-test three is the intersection of them, then it's reasonable this test does not result in good performance as its data is a intersection of the data from sub-test one and two.

For sub-test (4), this experiment shows that the new network gained the highest performance improvement among the four sub-tests. The new network can take advantage of the sentiment information in the non-subjective questions and their answers, and make a better prediction in the category of non-subjective questions with more sentiment contained.

As can be seen from Table II, non-subjective Questions with Sentiment have the highest performance gain within those four tests, and subjective Questions with Sentiment have the least performance gain. Further analysis of the original validation and testing dataset reveal that the validation set has higher numbers of non-subjective questions with sentiment (340), and has few of subjective questions with sentiment (77). The testing set has 299 questions and 99 questions in those categories. While subjective questions with sentiment is a subset of subjective questions, and its performance is worse than subjective questions. We can infer that subjective questions without sentiment have a higher performance than subjective questions. For this category, the validation set also a higher number of subjective questions without sentiment (449), while testing set has 379 questions in this category.

V.CONCLUSION AND FUTURE WORK

In this study, we demonstrated that adding sentiment information to the biLSTM/CNN [8] can improve the overall performance compared to the baseline on both Precision@1 and MRR evaluation measures. The Precision@1 performance gain reached 13.76% for the validation set, and 1.25% for the testing set. While the baseline model performance varies by 11.10% between validation set and testing set, the new model's performance is both higher and more stable, indicating the new model has a more consistent performance.

Based on the four sub-tests, we also conclude that the similarity-based answer ranking method works better for non-subjective questions. On the other hand, non-subjective questions with sentiment have the highest performance increase and P@1 after adding an extended representation of sentiment into the neural network. The difference in the question type distribution in validation set and test set is also one of the reasons why performance varies between them.

Potential future work includes finding a dataset with a higher ratio of subjective questions or questions with sentiment. The performance in those two categories could be improved since the model could extract more feature in those specific categories. If there are sentiment analysis framework could give sentiment information on an extended representation, and also categorize them into with sentiment/without sentiment. This practice could avoid the problem of having different judging criterion between different sentiment analysis framework for the sub-tests.

REFERENCES

[1] J.-H. Oh, K. Torisawa, C. Hashimoto, T. Kawada, S. D. Saeger, J. Kazama, and Y. Wang, "Why Question Answering using Sentiment Analysis and Word Classes," in *EMNLP-CoNLL '12 Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, 2012, pp. 368–378.

[2] A. Mishra and S. K. Jain, "Computing Sentiment Polarity of Opinion WHY Type Question for Intention Mining of Questioners in Question Answering Systems," Research in Computing Science, vol. 110, pp. 31–40, 2016.

[3] Heilman, M., & Smith, N. (n.d.). Good Question! Statistical Ranking for Question Generation. Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics, 609-617. Retrieved June, 2010.

[4] W.-tau Yih, M.-W. Chang, C. Meek, and A. Pastusiak, "Question Answering Using Enhanced Lexical Semantic Models," Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pp. 1744–1753, Aug. 2013.

[5] M. Wang, N. A. Smith, and T. Mitamura, "What is the Jeopardy Model? A Quasi-Synchronous Grammar for QA," Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 22–32, Jun. 2007.

[6] F. Eskandari, H. Shayestehmanesh, and S. Hashemi, "Predicting best answer using sentiment analysis in community question answering systems," 2015 Signal Processing and Intelligent Systems Conference (SPIS), 2015.

[7] M. Feng, B. Xiang, M. R. Glass, L. Wang, and B. Zhou, "Applying deep learning to answer selection: A study and an open task," 2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015.

[8] M. Tan, C. dos Santos, B. Xiang, and B. Zhou, "LSTM-based Deep Learning Models for Non-factoid Answer Selection," arXiv:1511.04108, 2016.

[9] L.-W. Ku, Y.-T. Liang, and H.-H. Chen, "Question Analysis and Answer Passage Retrieval for Opinion Question Answering Systems ," International Journal of Computational Linguistics & Chinese Language Processing, Volume 13, Number 3, September 2008: Special Issue on Selected Papers from ROCLING XIX, pp. 307–326, Sep. 2008.

[10] Jurafsky, D., & Martin, J. (n.d.). Speech and Language Processing (3rd ed.).

[11] C. Manning, M. Surdeanu, J. Bauer, J. Finkel, S. Bethard, and D. Mcclosky, "The Stanford CoreNLP Natural Language Processing Toolkit," Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, 2014.

[12] B. Felbo, A. Mislove, A. Søgaard, I. Rahwan, and S. Lehmann, "Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm," Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, 2017.

[13] S. Loria, P. Keen, M. Honnibal, R. Yankovsky, D. Karesh, and E. Dempsey, "Textblob: simplified text processing," Secondary TextBlob: Simplified Text Processing, 2014.

[14] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed Representations of Words and Phrases and their Compositionality," Advances in Neural Information Processing Systems 26 (NIPS 2013), pp. 3111–3119, Dec. 2013.