Question Classification Based on Hybrid Neural Networks

Zhongcheng Zhou[1, a], Xiang Zhu[2, b], Zhonghe He[3, c], YinchuanQu[4, d]

1College of Computer, National University of Defense Technology, China

2College of Computer, National University of Defense Technology, China

3College of Computer, National University of Defense Technology, China

4BeijingGaodiInformationTechnologyCo. Ltd., China

aceaserz@163.com, bzhuxiang@nudt.edu.cn,criterhe@outlook.com,dqyc@gaodig.com

Keywords: text classification, deep learning, CNN, LSTM

Abstract. Question classification is an important step in question answering system. There are many previous work based on machine learning about question classification. Although they are effective and practical, most of them require finding specific features to train the designed classifier. So it cannot be in common use in all the situations. Recently, deep learning methods has shown its remarkable strength in natural language processing. Deep neural networks like CNN and LSTM are helpful for sentence classification. The CNN is able to extract high-level local features while the LSTM can remember and discard information according to the context. Thus we implement a Hybrid Neural Network to handle different parts of a query. Both CNN and LSTM are utilized to achieve question classification. We conduct experiments comparing with methods based on machine learning and deep learning. The experimental results show the effectiveness and efficiency of our methods.

Introduction

The Question Answering (QA) system aims at automatically finding the answers to an arbitrary question in natural language. There are many successful QA systems such as IBM's Wason [1], WolframAlpha [2]and so on. The QA problem consists of several sub-problems, in which the question classification is included and it is important for question answering. The task of question classification is to predict the type of answer according to the query. In general, the question is described with title and description. As the questions in the question answering communities, such as Quora [3] and Yahoo Answers [4], the title of the question is typically short and straightforward while the description is usually longer and contains a lot of background information. Traditional question classification methods mostly are based on machine learning approach. For example, Naive Bayes, k-Nearest Neighbors and SVM algorithm can be used to implement the question classification. While the performance is dependent on the features used in the model. In most cases, bag-of-Words and bag-of-n-grams are the features utilized in machine learning approach. With the development of deep learning, neural network methods show the notable strength in Natural Language Processing (NLP). Though deep learning methods are firstly used in the field of computer vision, recent literature demonstrate its ability in text classification. The convolutional neural networks are capable of extracting the high-level features from local text by window filters. And the Long-Short Term Memory (LSTM) network is a special kind of Recurrent neural network (RNN), which is able to remember and ignore the features according to the context. Those characteristics are useful in question classification. In this paper, we combine those two kinds of neural networks into our framework to process different part of a question. And a Hybrid Neural Network (HNN) is implemented to achieve question classification.

Related Work

The Question Answering (QA) is a research activity which is related to Information Retrieval (IR) and NLP, but it is quite different from them in certain way. QA is a very practical activity and is

more like a collection of tools and techniques than formulas and theorems. In the first step of QA problem, it is important to classify the question according to its query. There are many previous work has been done to solve that problem [5]. Li et al. [6]present a machine learning approach to question classification. It learns a hierarchical classifier that is guided by a layered semantic hierarchy of answer types, and eventually classifies questions into predefined the classes. Huang et al. [7] propose head word feature and present two approach to augment semantic features of such head words using WordNet [8]. Mishra et al. [9] propose a method under the machine learning framework to achieve question classification. It combines lexical, syntactic and semantic features can be extracted from a question. A lot of efforts are paid in the field of QA. There is a QA track [10] in Text Retrieval Conference (TREC) since 1999. That task is to find the answer for a question in the open domain. It attracts researchers all over the world to participate in that evaluation every year and propose new method to solve the QA problem.

Deep learning methods play an important role in NLP tasks such as image recognition, sentiment analysis, text classification and so on. For example, LeNet-5 [11] is a convolutional neural network (CNN) designed for handwritten and printed character recognition. Like most other neural networks, it is trained with the back-propagation algorithm. The CNN has recently achieved remarkable performance on the task of sentence classification [12,13,14]. Kim Yoon [15] reports on a series of experiments with CNN trained on top of pre-trained word vector model for sentence-level classification tasks. The CNN utilize layers with convolving filters that are applied to local features. Even though the CNN is originally invented for computer vision, it is subsequently shown effective in NLP tasks. The experimental results show the advantages of CNN in text classification, including sentiment analysis and question classification. Zhang et al. [16,17] offer an empirical exploration on character-level convolutional neural networks for text classification. That model a sequence of encoded characters as input and the encoding is done by a predefined alphabet of size m. Different from word-based CNN, it quantize each character with 1-of-m encoding. So each character is transformed into a vector and its dimensionality is m. The results show that character-level CNN could work for text classification without the need of words. It indicates that language could be also thought of as a signal which is not different from other kind. Johnson et al. [18] propose a novel framework with CNN which is not rely on word embeddings. It learns embeddings of small text regions from unlabeled data for integration into a supervised CNN. The goal of region embedding learning is to map text regions into high-level concepts. While it is difficult for word embedding learning since individual word in isolation is insufficient to correspond to high-level concepts. From the previous literature, we can see that the CNN is capable of extracting high-level concepts or features from the text sequence.

The LSTM which is a special kind of RNN, is sensitive to the context. Lai et al. [19] apply a recurrent structure to capture contextual information as far as possible when learning word representations. It is able to introduce less noise compared to traditional window-based neural networks. Iyyer et al. [20] introduce a RNN model, QANTA, which learns word and phrase-level representations that combine across sentences to reason about entities. Encoding the intrinsic relations between the sentences in the semantic meaning of a document is a challenge in document-level sentiment classification. Tang et al. [21] introduce a RNN model to learn vector-based document representation. Zhou et al. [22] combine CNN and LSTM to achieve a novel and unified model called C-LSTM for sentence representation and text classification.

In recent years, the model based on deep neural networks demonstrate its strength in sentence representation and text classification. In this paper, we propose a framework based on CNN and LSTM to achieve question classification task.

Proposed Method

The framework of the Hybrid Neural Networks (HNN) is demonstrated as Fig. 1, It mainly consists of two components, a convolutional neural network (CNN) and a long-short term memory (LSTM) network. The CNN is utilized to extract the sequences of word features, which captures the local features of the text. And the LSTM uses the sequences of word features as input to capture

long-term dependencies over windowfeature sequences respectively. Furthermore, the query in QA system usually contains a title and a description. The title is a sentence which represent what user want to know. It is often simple and short. Unlike the title, the description depicts the question comprehensively, including the background and the related detail of the question. The CNN is good at extracting local features of the text, while the LSTM do well in the situation where the context is important. In the QA system, the title of the query is often short and unrelated to the context, so we can use several filters with different window size to capture the high-level features. After that, we extract the most important feature by a subsampling layer. On the other hand, the description of a query is usually long and the meaning of it is related to its context. And therefore, it is proper to use LSTM to tackle this problem. However, the LSTM is not able to capture the local features of the text. In order to overcome that dilemma, we utilize a CNN to extract the high-level features and recombine those features according to the time sequence. After that, those sequential features are taken as the input for the LSTM. The details of our framework are introduced in the following subsections.

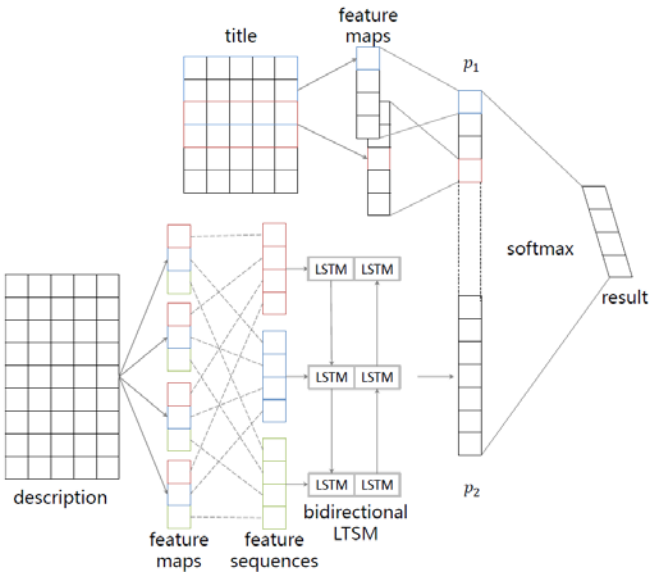

Fig. 1. The framework of hybrid neural networks

A. Feature Extraction through Convolution Neural Networks

In this subsection, we introduce how we capture the local features of the text by CNN. First of all, we select the word vector model as our language model to represent each word in the text. The word vector model is trained by external corpus, such as Wikipedia, Freebase and so on. A word2vec tool called genism[23]is utilized to generate the word vector model. After that, each word t in the text can be transformed into a vector according to the previous word vector model. For a word t, it can be represented ast $= (w_1, w_2, \cdots, w_N)$based on word vector model, where N is the dimensionality of the word vector model and $w_i$is the weight in the $i-$ thdimension. By this mean, a sentence can be transformed into a matrix.

Next, we make a toy example as Fig. 2 to demonstrate how the word vector model works in detail. In the Fig.2 the sentence s can be formulated as a word sequence s =Who $\oplus$ will $\oplus$ win $\oplus$ the $\oplus$ NBA $\oplus$ finals, where $\oplus$ is the concatenation symbol. Each word in the sentence is transformed into a vector, then the sentence can be transformed into a matrixA $\in R^{MN}$by the same way, where M is the length of the title.

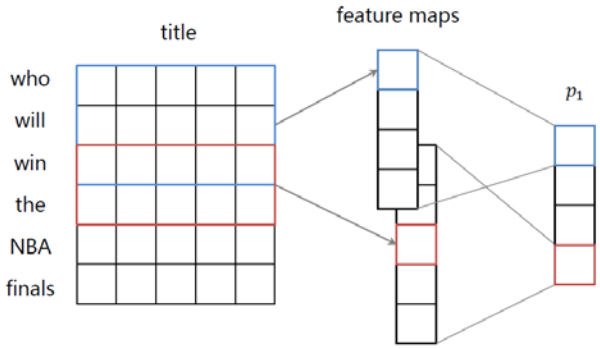

Fig. 2. The structure of the convolutional neural networks

After that, the matrix A is utilized as the input to train the CNN. The first layer is a convolutional layer with multiple width filters. Let $w \in R^{HN}$ denote the filter in the convolutional layer, where H is the width of the filter and N is the dimensionality of the word vector model. It is applied to a window word sequence of H and each word is represented by a vector of dimensionality N. For convenience, let $t_{i:i+H}$ denote the word concatenation $t_i \oplus t_{i+1} \oplus \cdots \oplus t_{i+H}$. When we apply the filter w to t_{i+H}, it will generate a new feature c_i. It can be formulated as follows,

$$c_i = f(w \otimes t_{i:i+H} + b). (1)$$

where $\otimes$ is the convolution symbol, b is a bias parameter and $f(\cdot)$ is a non-linear function. Then, we can slide the filter through the whole title to generate a feature map based on all the word concatenation t_{i+H}. As a result, the feature map ccan be formulated as follows,

$$c = (c_1, c_2, \cdots, c_{M-H+1})^T (2)$$

where $c \in R^{M-H+1}$ is the feature map and M is the length of the title. Note that, we make the length of the filter equal to the dimensionality of the word vector model in this paper, because it is meaningless to generate a feature with a filter whose length is less than the dimensionality of a word vector. After that, a subsampling layer is utilized to make a pooling operation to feature map c. By the pooling operation, the maximum value c_{max} in the c will be computed as $c_{max} = \max c_i \, (c_i \in c)$. The maximum value c_{max} reflects the mostsignificant feature in the feature map which is generated by a certain filter. The subsampling layer can help capture the most important semantic feature in the text and can also avoid overfitting problem. By this means, we can compute all the c_{max} for the filters of different length H. Then all the c_{max} are combined as a vector p_1. As this point, we extract the valuable features from the title of a query through a CNN and those features are formulated as a vector p_1. It is useful for classification in the next step.

B. Long-Short Term Memory Networks

In the previous subsection, we introduce how we use CNN to process the title of the query. Next, we demonstrate the approach of processing the description of the query. Because LSTM is insufficient to capture the local features of the text, a CNN is utilized to extract the high-level features in the first step. In that step, several filters with the same width are used to extract different features. By different filters, we can get different local feature maps. Those feature maps reflect the semantic characteristics from different perspective. After that, we rearrange the feature maps according to the time sequence. Then we can get feature sequences as the input for the LSTM and each feature sequence consists of several features generated by different filters. The feature sequence reflects the local features from different view.

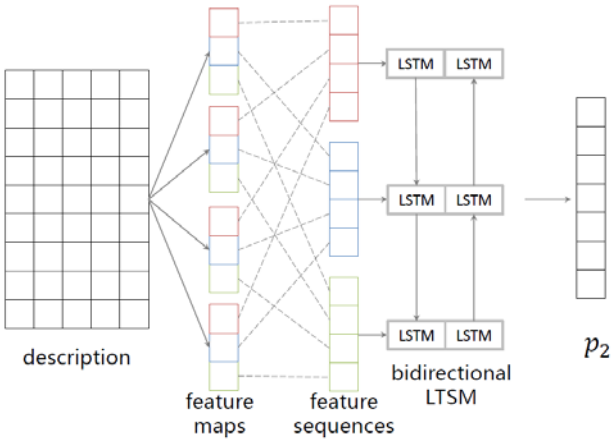

Fig. 3. The structure of the BiLSTM networks

We give an example to illustrate the process as Fig. Let s' denote the description in the query and the length of s' is M'. And $W = \{w_1, w_2, \cdots, w_{L'}\}$represents the filter set. The size of W is L' and the width of the filter $w_i \in$ Wis H'. After that, when we slide the window through the description in the query and make convolution operation with the filter w_i as Eq. (1), a feature map $c_i$will be generated. The feature map $c_i$can be formulated as follows,

$$c_i = (c^{i,1}, c^{i,2}, \cdots, c^{i,M'-H'+1})^T \quad (3)$$

Then in order to get the different local features, we rearrange the feature maps with respect to the time sequence. After rearrangement we can get feature sequences and let c'_j denote a feature sequence. It can be formulated as follows,

$$c'_j = (c^{1,j}, c^{2,j}, \cdots, c^{L',j})^T \quad (4)$$

where $j \in [1, M' - H' + 1]$. By this way, we rearrange the feature maps into the feature sequences. Then those feature sequences are utilized as the input for a bidirectional LSTM. In most cases, users describe their problem with much background introduction. While the traditional LSTM can only record one-way information, the bidirectional LSTM can record the two-way information. It is useful for deciding which part of text can be discarded and which part of text can be retained according to the context. So the bidirectional LSTM is a good choice for this problem. When we apply bidirectional LSTM to the feature sequences, it will generate new features. We denote the output of bidirectional LSTM p_2. Then we combine the features p_1, which is generated by CNN, and p_2 as the final features p. At last, the final features p acts as the input of a fully connected softmax layer. The output of the layer is the query classification distribution over all the categories.

Experimental and Analysis

In this section, we run several experiments to verify the proposed method. Firstly, we introduce our experimental settings. Then, we demonstrate the experimental results in detail and make an analysis of the results. Our experiments are conducted on a machine with 4 cores (Intel Xeon 1.80GHz CPU) and 64 GB memory, running 64bit Ubuntu 16.04. All the algorithms compared in the experiments are implemented with Python 3.5 and Keras 1.0.7.

Table.1. Dataset characteristics

Name	TrainingSet #	TestSet #	Category #
Baidu Knows	104922	20985	14
TREC	206417	49868	26

A. Experimental Settings

Datasets. Table. 1 demonstrates the characteristics of the dataset used in our experiments. The Baidu Knows and TREC are the benchmarks in the literature of text classification. The datasets are of different topics and characteristics. The Baidu Knows is a Chinese language collaborative Web-based collective intelligence by question and answer provided by the Chinese search engine Baidu. The TREC is a popular dataset in TREC LiveQA Track [24], most of the questions and answers are the real data in the social platform Yahoo! Answers.

Algorithms. We compare our method with other benchmark methods, SVM, MaxEnt, CNN, LSTM and BiLSTM. The SVM is a supervised learning model with associated learning algorithms, it is widely used in classification and regression analysis. The MaxEnt classifier is a classification method that generalizes logistic regression to multiclass problems. The CNN, LSTM and BiLSTM are the models mentioned in Section. III, we run the title and description of a query in single model to make comparison.

B. Results and Analysis

In this subsection, we select classification precision, recall rates and F1 score as measurements. For each dataset, we run every algorithm on it by 5-fold cross-validation and use the average performance as the experimental results. The experimental results are demonstrated as follows.

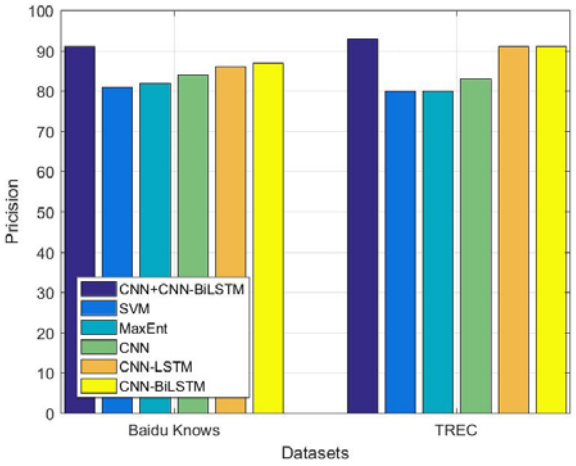

Fig. 4. The precision performance comparison

Firstly, the Fig.4 shows the precision performance of all the algorithms. We can see that our method (CNN+CNN-BiLSTM) outperforms other algorithms with respect to precision rates. Furthermore, the CNN-BiLSTM method gets the second-best performance. That is because the CNN+CNN-BiLSTM method is a hybrid method that using CNN to process query title and CNN+BiLSTM to handle the query description. The CNN is good at extracting local features of the text, while the query title usually is a short text, so it is proper to use the CNN to process query title. Furthermore, the BiLSTM do well in the situation where the context is important and the query description is a passage, so the CNN-BiLSTM has a good performance.

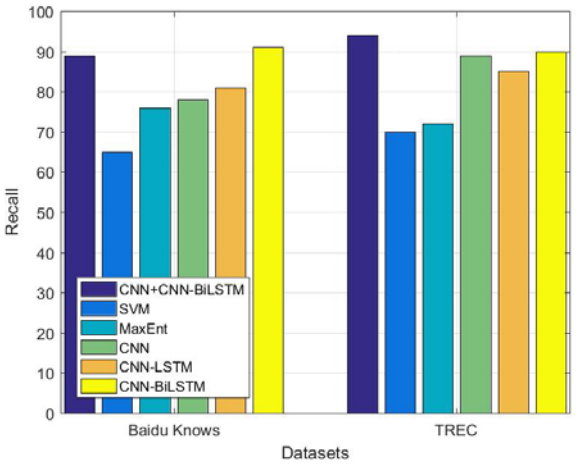

Fig. 5. The recall performance comparison

Then, the Fig.5 shows the recall performance of all the algorithms. We can see that our method (CNN+CNN-BiLSTM) has a similar recall performance to the performance of CNNLSTM and CNN-BiLSTM, and outperforms other methods.That is because LSTM is sensitive to the context and is able to introduce less noise compared to traditional window-based neural networks. So it has a high recall rates.

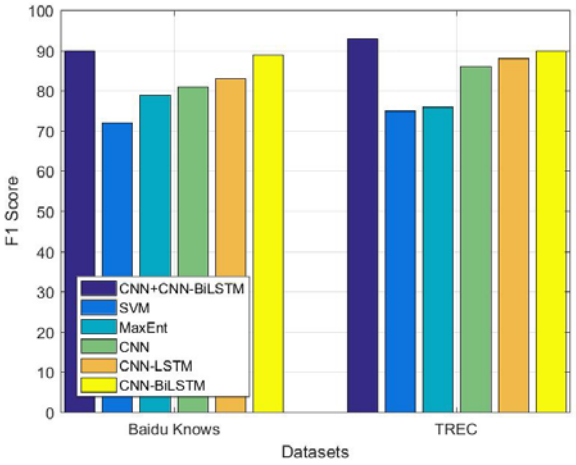

Fig. 6. The F1 performance comparison

Next, the Fig.6 shows the F_1 score performance of all the algorithms. We can see that our method (CNN+CNN-BiLSTM) gets the highest average F_1 score, that means our method outperforms other methods when taking both classification precision and recall into consideration.

From the experimental results, we can see that our method can enhance the performance by utilizing proper model according to the text feature in different part of a query.

Conclusion

Question classification is an important step in question answering system. There are many effective and practical previous method to solve that problem, but most of them require finding specific features to train the classifier. Recently, deep learning methods has shown its remarkable strength in natural language processing. According to the text features in different part of a query, we use different model to process it. We implement a Hybrid Neural Networks that utilizing CNN to process the query title and BiLSTM to process the query description. We conduct experiments comparing with methods based on machine learning and deep learning. The experimental results show the performance of our methods.

Acknowledgements

The work is supported by National Basic Research and Development Program (No.2013CB329601,2013CB329604) and National Natural Science Foundation of China (No.61502517,No.61472433,No.61372191,No.61572492).

References

[1] D. Ferrucci, E. Brown, J. Chu-Carroll, J. Fan, D. Gondek, A. A. Kalyanpur, A. Lally, J. W. Murdock, E. Nyberg, J. Prager *et al.*:*Building watson: An overview of the deepqa project*. AI magazine. Forum vol. 31, no. 3(2010), p. 59–79.

[2] Information on https://www.wolframalpha.com

[3] Information on https://www.quora.com

[4] Information on https://answers.yahoo.com

[5] H. Sun, H. Ma, W.-t. Yih, C.-T. Tsai, J. Liu, and M.-W. Chang: *Open domain question answering via semantic enrichment*, in Proceedings of the 24th International Conference on World Wide Web. ACM(2015), p. 1045–1055.

[6] X. Li and D. Roth: *Learning question classifiers: the role of semantic information*, Natural Language Engineering, vol. 12, no. 03(2006), p. 229–249.

[7] Z. Huang, M. Thint, and Z. Qin: *Question classification using head words and their hypernyms*, in Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics(2008), p. 927–936.

[8] http://wordnet.princeton.edu

[9] M. Mishra, V. K. Mishra, and H. Sharma: *Question classification using semantic, syntactic and lexical features*, International Journal of Web & Semantic Technology, vol. 4, no. 3(2013), p. 39.

[10] E. M. Voorhees et al.: *The trec-8 question answering track report*. in Trec, vol. 99(1999), p. 77–82.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner:*Gradient-based learning applied to document recognition*, Proceedings of the IEEE, vol. 86, no. 11(1998), p. 2278–2324.

[12] N. Kalchbrenner, E. Grefenstette, and P. Blunsom: *A convolutional neural network for modelling sentences*, arXiv preprint arXiv:1404.2188.(2014)

[13] R. Johnson and T. Zhang: *Effective use of word order for text categorization with convolutional neural networks*, arXiv preprint arXiv:1412.1058.(2014)

[14] Y. Zhang and B. Wallace: *A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification*, arXiv preprint arXiv:1510.03820. (2015)

[15] Y. Kim:*Convolutional neural networks for sentence classification*, arXiv preprint arXiv:1408.5882.(2014)

[16] X. Zhang, J. Zhao, and Y. LeCun:*Character-level convolutional networks for text classification*, in Advances in Neural Information Processing Systems(2015), p. 649–657.

[17] X. Zhang and Y. LeCun:*Text understanding from scratch*, arXiv preprint arXiv:1502.01710. (2015)

[18] R. Johnson and T. Zhang:*Semi-supervised convolutional neural networks for text categorization via region embedding*, in Advances in neural information processing systems(2015), p. 919–927.

[19] S. Lai, L. Xu, K. Liu, and J. Zhao:*Recurrent convolutional neural networks for text classification*. in AAAI (2015), p. 2267–2273.

[20] M. Iyyer, J. L. Boyd-Graber, L. M. B. Claudino, R. Socher, and H. Daume III: *A neural network for factoid question answering over´ paragraphs*. in EMNLP(2014), p. 633–644.

[21] D. Tang, B. Qin, and T. Liu: *Document modeling with gated recurrent neural network for sentiment classification*, in Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing(2015), p. 1422–1432.

[22] C. Zhou, C. Sun, Z. Liu, and F. Lau: *A c-lstm neural network for text classification*,arXiv preprint arXiv:1511.08630.(2015)

[23] Information on http://radimrehurek.com/gensim/index.html

[24] E. Agichtein, D. Carmel, D. Harman, D. Pelleg, and Y. Pinter: *Overview of the trec 2015 liveqa track*, in The Twenty-Fourth Text REtrieval Conference (TREC 2015) Proceedings. National Institute of Standards and Technology (NIST).(2015)