Combining Lexical and Semantic-based Features for Answer Sentence Selection

Jing Shi[a], **Jiaming Xu**[a,*], **Yiqun Yao**[a], **Suncong Zheng**[a], **Bo Xu**[a,b]
[a]Institute of Automation, Chinese Academy of Sciences (CAS). Beijing, China
[b]Center for Excellence in Brain Science and Intelligence Technology, CAS. China
{shijing2014, jiaming.xu, yaoyiqun2014}@ia.ac.cn
{suncong.zheng, xubo}@ia.ac.cn

Abstract

Question answering is always an attractive and challenging task in natural language processing area. There are some open domain question answering systems, such as IBM Waston, which take the unstructured text data as input, in some ways of humanlike thinking process and a mode of artificial intelligence. At the conference on Natural Language Processing and Chinese Computing (NLPCC) 2016, China Computer Federation hosted a shared task evaluation about Open Domain Question Answering. We achieve the 2nd place at the document-based subtask. In this paper, we present our solution, which consists of feature engineering in lexical and semantic aspects and model training methods. As the result of the evaluation shows, our solution provides a valuable and brief model which could be used in modelling question answering or sentence semantic relevance. We hope our solution would contribute to this vast and significant task with some heuristic thinking.

1 Introduction

Selection-based question answering (QA) is a task in question answering to pick out one or several parts in a context containing an answer to an open-domain question, where the context comprises of one or more sentences. Commonly, a typical pipeline of open-domain question answering systems is composed of three high level major steps: a) question analysis and retrieval of candidate passages; b) ranking and selecting of passages which contain the answer; and optionally c) extracting and verifying the answer (Prager, 2006; Ferrucci, 2012). In this paper, we pay close attention to the answer sentence selection. Being considered as a key subtask of QA, the selection is to identify the answer-bearing sentences from all candidate sentences. The selected sentences should be relevant to and answer the input questions (Wang and Nyberg, 2015). Several corpora have been created for these tasks like TREC-QA , WikiQA (Wang et al., 2007; Yang et al., 2015), allowing researchers to build effective question answering systems (Voorhees and others, 1999; Andreas et al., 2016; Dai et al., 2016; Yih et al., 2014; Yu et al., 2014; Zhang et al., 2016).

The nature of this task is to match not only the words but also the meaning between question and answer sentences. For example, the answer to "Where was James born ?" is more likely to be "He came from New York ." than "James was born in summer." , even though the latter is more similar in the superficial level. Further, the crisis of the task is to find the sentence most closely related to the intention of the question.

There have been many works towards the sentence selection task (Heilman and Smith, 2010; Wang and Nyberg, 2015; Wang and Manning, 2010; Severyn and Moschitti, 2013). Basicly, those models could be divided into two categories: the lexical models and semantic-based models. The relatedness between the question-answer sentence pair measured by lexical models is mostly based on some metrics such as Longest common substring (LCS), Bag-of-Words (BOW) and Word Overlap Ratio as well as

*Corresponding author.

Proceedings of the Open Knowledge Base and Question Answering (OKBQA) Workshop,
pages 30–38, Osaka, Japan, December 11 2016.

Table 1: Some samples in the dataset. The identifier "\t" splits each line into 3 parts: the question, the candidate answer and the label, where 0 is incorrect while 1 is the right answer.

蜓蜥属在哪里有分布？ \t 中文名: 股鳞蜓蜥 \t 0
蜓蜥属在哪里有分布？ \t 俗名别名: \t 0
蜓蜥属在哪里有分布？ \t 英文名: SouthChina forest skink \t 0
蜓蜥属在哪里有分布？ \t 拉丁学名: Sphenomorphus incognitus \t 0
蜓蜥属在哪里有分布？ \t 地理分布: 分布在台湾南部与东部。 \t 1

Table 2: Statistics of the training dataset. Each pair denotes a question-candidate answer pair. Average Pairs is the average number of pairs in one question.One2One means the question only has one answer while One2Many means at least 2 answers.

Questions	Pairs	Average Pairs	One2One	One2Many	Positive pairs	Positive %
8772	181,882	20.73	8,459	313	9,198	5.06

some complex syntactic matching degree. The semantic-based models usually use some neural network framework to obtain the distributed representation between the sentences . However, both the two categories get some disadvantages. The former could just capture the similarity in literal level , losing sight of the deep semantic information and latent correlation; Meanwhile, the semantic-based models often take much time to train and rely heavily on the provided data. When the train dataset is insufficient or there are some unseen works in test phase, the performance is hard to guarantee.

To solve those problems, we present a model that emphasizes the intention analysis of the question through a feature engineering method. The critical part of the model is to build some efficient lexical features integrated with semantic-based methods to measure the relevance between Chinese question and the answering sentences. Our contributions are three-fold:

- We propose a supervised approach by combining lexical and semantic features to solve the sentence selection task in open-domain QA.
- We explore a feature named Intention Analysis Window Feature which can flexibly construct a strong semantic relation between question and answer sentences. The feature is also capable of integrating kinds of external resources, which could reinforce the performance and effectiveness.
- An efficient Topic Word Extraction method is exploited in our model to successfully filter irrelevant information in answer sentence selection process.

Our model is simple, low-cost in computation and commonly adaptive to various questions. As the result of the evaluation completion shows, the full model is highly efficient, outperforming almost all other models except one with external knowledge resources.

2 Corpus and Problem Description

The aim of common sentence selection task is to choose one or more sentences from the candidate lists to answer the question. At the conference on Natural Language Processing and Chinese Computing (NLPCC) 2016, China Computer Federation, along with the Microsoft Research Asia, organized a shared task evaluation about Open Domain Question Answering in Chinese. One question from the provided dataset in this evaluation is as illustrated in Table 1: each question has a sentences list from which to choose the answer or answers and each question alone with one sentence from its list form a question-candidate answer pair. In the training dataset, the label of every pair has been provided: 1 for right answer sentence and 0 for not. Table 2 shows the statistics of the training data. Most of the questions have only one answer and on average each question has 20.73 sentences from which to choose the answer.

Based on the form of sentence pair in the dataset, we could naturally use the sentence relevance within the pair to classify it. By constructing some suitable features, we take each pair as one sample to

Table 3: Some samples to show the entities close to the interrogative word. The bold words are the Interrogative word while the words with underline show the near entities.

Interrogative word	Samples
谁	电视剧《枪花》中的两大“枪花”分别由**谁**扮演？
什么	楚姓主要的来源是**什么**？
多少	型护卫舰可容纳**多少**人？
哪里	许地山早期代表作《缀网劳蛛》在**哪里**发表的？

train a binary classifier through machine learning method. Finally, we use the score of the 0-1 classification result, which is also the possibility of positive label, as the final score to calculate the MRR and MAP result by the official evaluation script.

3 Approachs

In this section, we describe the approach adopted by us in detail. As mentioned above, the whole model is a binary classification problem according to the relevance between question-candidate answer pair. The main content of this section can be summed up in two aspects: features and training. Section 3.1 contains a detailed description of our Intention Analysis Window Feature. Section 3.2 describes an important preprocessing method adaptive to the answer sentence selection task. And Section 3.3 contains the machine learning model and tool we choose to train our model. It should be pointed out that we use the jieba[1] tools for Chinese text segmentation.

3.1 Feature Description

After analyzing the dataset, we get many question-candidate answer pairs. Startring from basic idea, we can take each pair as an independent sample, then construct features from both literal and semantic aspects. However, we find the fact by experiments that it is quite a rude method to take each pair as an isolated sample. Because it just constructs samples independently from each pair, without considering the differences between various questions. In other words, besides the relation between the two sentences, whether a question-candidate answer pair could be positive label should also be considered synthetically under the whole answer lists of this question. As a result, we design our features under the consideration of the contextual environment.

3.1.1 Intention Analysis Window Feature.

Intention Analysis Window Feature (dubbed IAWF) is a method to get the vectorization representation of the relevance between question-candidate answer sentences pair by making full use of the question intention. This method is quite simple and efficient, and universal to kinds of different questions. In our experiments, this method results in an obvious improvement over the performance of our model.

Most often, during the pipeline of conventional QA system, question analysis is an important step. The aim of this step is to analyze and comprehend the intention, and then to assist in subsequent retrieval and answer extraction. Through a careful observation of the dataset, we find there is roughly a rule that entity closer to the interrogative word covers more semantic information to represent the sentence. As some examples illustrated in Table 3, the entities could properly express the key information of the sentences, especially when given the corresponding answer lists of the question.

Based on this observation, we design an algorithm to make fully use of this characteristics. The whole process is showed in Figure 1. To a question sentence q and a candidate answer sentence x, we first get word segmentation with PosTag (Part-Of-Speech Tag) of q and identify the location of the interrogative word. Then we choose the entities, which have a distance of 1,2,3 to the interrogative word. Each distance is bi-directional, and the entity with a distance beyond the range of the sentence will be set as ‘None’. Here we refer the entity to the word whose POS (Part-Of-Speech) is noun or verb. In this way, we get three groups of entities, each of the group has two entities and one of them maybe ‘None’. To each entity in each group, we calculate its relevance score to every candidate answer sentence *sent*

[1] https://github.com/fxsjy/jieba

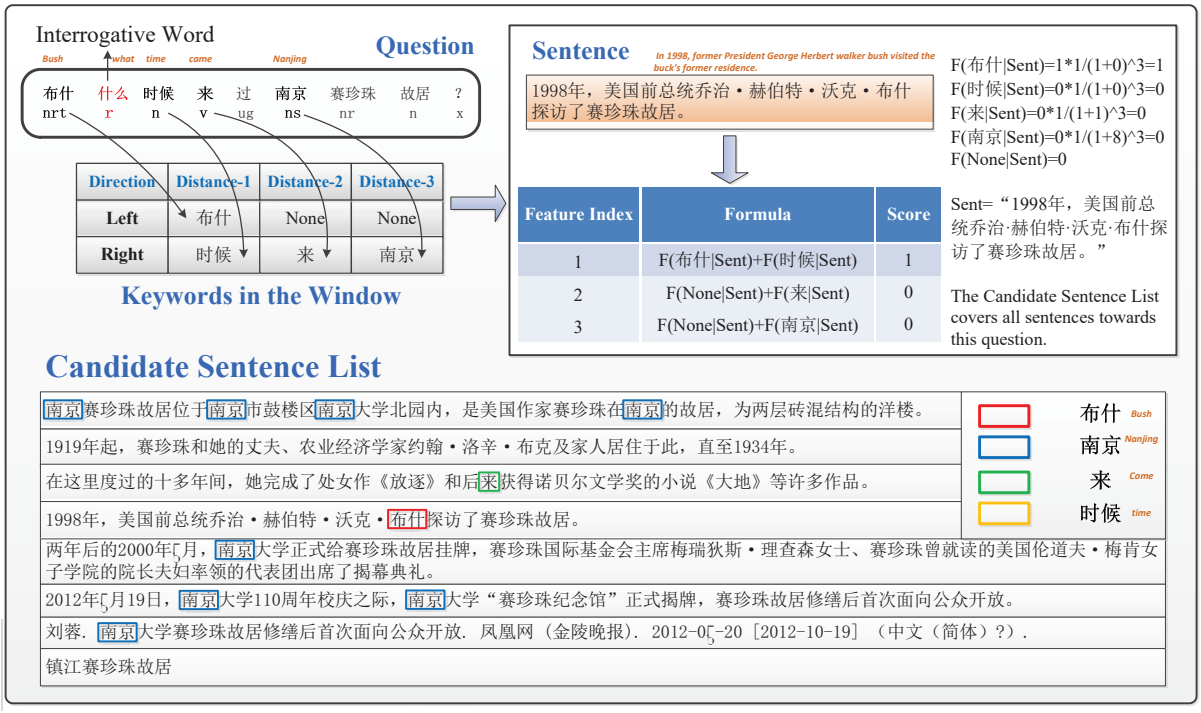

Figure 1: The whole process to extract Intention Analysis Windows Feature with question as “布什什么时候来过南京赛珍珠故居？”. The answer list of this question is showed at the bottom. We choose the 4th sentence in the answer list to serve as example. Note that we omit the term of Passage in equation to calculate each score.

according to an idea like tf-idf. To be specific, the score of each *word* to one sentence *sent* in the candidate sentences list *Passage* is:

$$F(word|Sent, Passage) = \begin{cases} \dfrac{sgn(G(word|Sent))}{\sum\limits_{\{s|s\in Passage, s\neq Sent\}} (1+G(word|s))^3} & else \\ 0 & if \quad word =' None' \end{cases}$$

Where, $G(word|s)$ means the times of word appearing in sentence s, $sgn(t)$ is sign function, $sgn(t) = 1$ when $t > 0$ and $sgn(t) = 0$ when $t = 0$.

At last, we add up the score in each groups to get three final score to be served as three features of each question-answer pair.

To summarize, Intention Analysis Window Feature is to choose some entities close to the interrogative word from the question, and then calculate the score of the entities in each candidate answer sentence to measure the relevance degree between the question and answer sentences. If there is an entity from the question has showed only in one of the answer lists, then that sentence should get a high score. If an entity shows everywhere in answer lists, then that entity is mostly an unvalued word.

Of course, the window width of 3 or the index in above equation is not a fixed value. But during the experiments, we find it a suitable choice. Actually, as the average length of the question sentence is within 10, we could easily to lengthen the width of the window to cover more entities in the question sentence even the whole ones. However, the following effect is not good enough. The width of 2 is close to 3, but the width of 4 or more decreases apparently. We think that distant entities bring much more noisy than the beneficial information they cover.

3.1.2 Extension of IAWF

Actually, the basic Intention Analysis Window Feature construct a tight correlation between the question and answer sentences by making fully use of the keyword in the question. However, the critical word from the question sometimes doesn't exist in the answer sentences but be replaced by a highly relevant other word. For instance, for the question " Which season did the ACL 2016 hold? ", the IAWF keyword "season" doesn't appear in answer sentence "ACL 2016 held in summer ."

As a matter of this fact, we make some efforts to extend the IAWF with much more semantic integration. In detail, after getting the important entity , we could freely to import some external resources such as synonyms thesaurus or word2vec (Mikolov and Dean, 2013). The synonym word is the candidate sentence could be roughly considered as the same word of the keyword while the most similar word calculated from the word2vec could also be regarded as a variety of the keyword .Then the equation used in IAWF could be used, with a discount respectively, to get another group of features to model the pair.

In our experiments, the extension of IAWF could handle sorts of questions covering varietal word of the important entity in answer sentence. After extension of the IAWF, the model becomes capable to develop with integration of different resources, result in a wider adaptability.

3.2 Topic Word Extraction

In this part, we describe a very useful trick as a preprocessing method of the dataset. Just like the thought of IDF (Inverse Document Frequency), we find the topic word within the question often has a bad impact on choosing the right answer. For instance, the subject of the question maybe the alias of the topic word about the answer lists, which has showed just once. Then the score of this sentence covering the alias word is very high. However, the subject of the question is usually unvalued to analyze the intension of the question because the whole answer lists are its description.

To tackle this common problem, we manage to extract the topic word off the question sentence by some simple rules. The main rule is to recognize the topic word from the candidate sentences list by some patterns. For example ,the name of one people or place at the beginning of the list usually could be judged as the topic word. This method brings about 3% performance improvement in our test and increase the robustness during the cross validation process.

3.3 Training Model

We have considered some mainstream machine learning model serves as the classifier, including L-ogistic Regression, SVM (Cortes and Vapnik, 1995), Random Forest (Breiman, 2001), GBM (Friedman, 2001) and XGBoost (Chen and Guestrin, 2016). After referring to some papers (Joachims, 2002; Liu et al., 2016) and doing some simple comparison experiments, we found the XGBoost model almost reached the optimal performance. Furthermore, it is easy to merge with our features processing framework and fast enough. Finally, we choose XGBoost tree model as our classifier. There are some parameters could be adjusted in the XGBoost. Our choice is detailed in the Section 4.

4 Experiments and Evaluation

There are totally 8772 questions in the training dataset of this task, and each of the question-answer pairs has a handcrafted label. To evaluate our model, we divide the 8772 questions after shuffle into training ones and test ones with a ratio of 7:3. And we made 3 pairs of this training-test dataset to evaluate our model with some cross validation method. Besides the Intension Analysis Window Feature, we also build some conventional features to contrast and work together.

4.1 Basic Features

The NLPCC 2016 committee gives 4 baselines result of the train dataset: Average Word Embedding， Word Overlap， Machine Translation and Paraphrase. Further, there are 3 types of features we used in our work.

Verbatim Features. We construct the verbatim features from the literal similarity between question and candidate answer. Simply, we use metrics as follows:

- **Longest common substring.** Longest common substring (LCS) method is a conventional metric widely used in language processing. In this task, we think the length of LCS could reflect the similarity at literal level between the two sentences. Besides, we take the ration of length of LCS to the length of the question as another feature in addition to length of LCS. It could to some extend increase the robustness of this metric.

- **Word overlap.** The same words in question and candidate answer sentence is a clue to find the answer. So we take the times that one word both in two sentences as another metric. Similarly, the ration of word overlap times to total word number in question is also added.

Bag-of-Words Features. Bag-of-Words (BOW) is a common idea in the language model, which is mainly used as a tool of feature generation. After transforming the text into corresponding vector, we can calculate various measures to characterize the text. In our task, the two sentences in each pair could be mixed to form a bag, then the sentences could be vectorized through the bag, making it available to calculate kinds of distance by various mathematical methods. For example, assuming one sentence is " 我\爱\你\大地\母亲\ 。 " while another sentence is "我\爱\你\山川\河流\。 ", then the bag of words will be [我\ 爱\ 你\ 大地\ 母亲\ 。 \ 山川\ 河流]. Following the words order in this bag, the vector of first sentence is [1 1 1 1 1 1 0 0], and the vector of another sentence is [1 1 1 0 0 1 1 1]. The 1 or 0 means the word in words bag is in this sentence or not. Both vector has a dimension of 8, same as the length of the bag. With this method, we construct the vecotors pair according to each question-candidate answer pair. After that, we calculate a series of distance between the two vectors in a pair such as: Cosine distance[2], Jaccard distance[3], Hamming distance[4] and City Block Distance[5]. Each of the results above serves as one dimension in the whole features of a question-answer pair.

Word Embedding Features. It is necessary to consider some suitable features to construct the relevance at semantic level. Naturally, we can use the word embedding trained from large scale corpus to model our sentences. Word2vec (Mikolov and Dean, 2013) vectors, size of 11428967, trained from Baidu baike[6] items are used. Each of the vectors has a dimension of 100. We construct the sentence representation as the average embedding of the words within it. Of course, there are many out of vocabulary words in our task dataset, so we initialize those words to a random 100-dimension vector respectively from Gaussian distribution with mean = 0 and $\sigma = 0.1$. Though the average embedding and random initialization contains some irrationality, for a multiple features engineering problem, each feature can exist a certain amount of imperfection, in the perspective of training it will be automatically measured with a trade-off. After getting the word2vec representation of the question and candidate answer sentences, we use Euclidean, Cosine, Jaccard, Hamming and City Block distances to calculate the similarity. Each of the results above serves as one dimension in the whole features of a question-candidate answer pair.

4.2 Results

The main results of our solution and official baselines are showed in Table 4. We contrast our model with 5 different forms: a) Basic model contains the features from the Verbatim Features, Bag of Words Features and Word Embedding Features. b) IAWF model contains the Intention Analysis Window Features. c) Mix model has the features from both Basic and IAWF models. d) Extension Mix model have the IAWF along with the use of synonyms thesaurus and the features from the basic Model. e) Extraction Mix+Extraction model adds the Topic Word Extraction method based on the Extension Mix model. The features used in each form are simply concatenated to form a full feature vector. It is worthy to know that the dataset of the task is a typical unbalanced one which has too much negative samples than

[2] https://en.wikipedia.org/wiki/Cosine_similarity
[3] https://en.wikipedia.org/wiki/Jaccard_index
[4] https://en.wikipedia.org/wiki/Hamming_distance
[5] https://en.wikipedia.org/wiki/Taxicab_geometry
[6] http://baike.baidu.com/

Table 4: The evalutaion results of some baseline method and our soulutions. ACC means the binary classification accuracy.

Model	Method	ACC	MAP	MRR
Baseline Models	Average Word Embedding	–	0.4598	0.4601
	Word Overlap	–	0.5105	0.5123
	Machine Translation	–	0.2408	0.2409
	Paraphrase	–	0.4876	0.4892
Our Models	Basic	0.9475	0.5482	0.5494
	IAWF	0.9616	0.6258	0.6279
	Mix	0.9641	0.7392	0.7409
	Extension Mix	0.9645	0.7652	0.7666
	Extension Mix+Extraction	**0.9654**	**0.7883**	**0.7901**

Table 5: Parameters of the XGBoost tree model.

max_depth	eta	min_child_weight	max_delta_step	Subsample	objective
7	0.06	80	50	1	binary:logsitic

the positive ones. As a result, the classifier tends to predict the negative label because it is easy to get a high classification accuracy. And almost the same accuracy may correspond a big difference in MAP or MRR.

Compared to the Basic model, Intention Analysis Window Features gains a great performance about 10% higher than the ensemble features from Verbatim Features, Bag of Words Features and Word Embedding Features. This demonstrates the effectivity of the Intention Analysis Window Features. Further, the extension of the IAWF brings a obvious promotion of MAP and MRR though the Mix model has already gotten almost 20 features. Besides, the method of Topic Word Extraction promotes the final result and shows a better robustness in cross validation. Finally, training our model with the whole given training dataset, under the best features and parameters, we get the performance of MAP=0.8263 and MRR=0.8269 in the test dataset given by the NLPCC official results.

To be specific about the training model, the final parameters of the XGBoost tree model we used is set as the Table 5. We find that the parameter tuning is an important process to affect the final metric. However, the parameters with a reasonable range are easy to find after a few attempts. Parameters with reasonable range could almost reach limit of the features and plenty of fine tuning could at most affect the MAP or MRR by only 1.5%. That is to say, parameter tuning should not be regarded as the key point of the system and the features themselves are the critical factor.

At the beginning of the process to construct the features, we just do experiments on one training-test dataset and with the same parameters of XGBoost. Because the upside potential of the feature engineering is quite large and far away from the limit. And the best parameters are gained from the result of cross validation at the final phase of the feature engineering.

5 Discussion

From the whole process of construction, adjustment and experiments, we get some intuition and experience within the sentence selection task.

Firstly, the candidate sentences list is crucial to the success of the question. As the promotion of result brought by the IAWF model shows, the isolate basic features couldn't catch the specificity of every candidate sentence list. Only by analyzing the question intension under the environment of context could the purpose be extracted correctly. So, under the consideration of a fine traditional QA system, retrieval of candidate passages or sentences is of much importance before the sentence selection.

Secondly, from the proper functioning of the IAWF models, we can draw a impression that the syntactic construction of the answer sentence has very litter impact on the analysis of the intension. Because the algorithm we use take the answer sentences as a unordered bag rather than a sequence. So

we infer with a bit radicalness that the Recurrent Neural Networks (RNN), which focus on sequential information, maybe not a proper choice. As far as we know, the best result of the evaluation solution also choose the framework of Convolutional Neural Network (CNN).

Finally, from feature engineering's point of view, the IAWF gets great progress, more than 10% in specific, after mixing with the basic literal features. This phenomenon means that these two groups of features is highly complementary, completing different functions in this task. Unlike the LCS or Word Overlap features, the IAWF gives much attention about the individual keyword within the QA process rather than the similarity between two whole sentences.

However, there is still much work to do. Our model is still unable to exactly handle some question whose purpose is to choose a subclass of the keywork. And we just test our approach in Chinese QA, other languages also need to be examined to find out whether this method or some conclusion is fit to general language phenomenon or just chinese Characteristics.

6 Conclusion

This paper presents the solution of our model with feature engineering in open-domain document based question answering task at NLPCC 2016 conference. In our model, the combination of some conventional and original, lexical and semantic-based features along with useful extraction method is employed to construct feature groups for the question answering pairs. Our solution can be successfully conducted with a high speed and at very low computation cost. The results show that our model is simple and efficient.

Acknowledgements

We thank the anonymous reviewers for their insightful comments, and this work was supported by the Strategic Priority Research Program of the Chinese Academy of Sciences (Grant No. XDB02070005), the National High Technology Research and Development Program of China (863 Program) (Grant No. 2015AA015402) and the National Natural Science Foundation (Grant No. 61602479).

References

Jacob Andreas, Marcus Rohrbach, Trevor Darrell, and Dan Klein. 2016. Learning to compose neural networks for question answering. In *NAACL*.

Leo Breiman. 2001. Random forests. *Machine Learning*, 45(1):5–32.

Tianqi Chen and Carlos Guestrin. 2016. Xgboost: A scalable tree boosting system. In *KDD*. ACM.

Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning*, 20(3):273–297.

Zihang Dai, Lei Li, and Wei Xu. 2016. Cfo: Conditional focused neural question answering with large-scale knowledge bases. In *ACL*.

David A Ferrucci. 2012. Introduction to this is watson. *Ibm Journal of Research and Development*, 56.

Jerome H Friedman. 2001. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232.

Michael Heilman and Noah A Smith. 2010. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions.

Thorsten Joachims. 2002. *Learning to classify text using support vector machines: Methods, theory and algorithms*. Kluwer Academic Publishers.

Guimei Liu, Tam T Nguyen, Gang Zhao, Wei Zha, Jianbo Yang, Jianneng Cao, Min Wu, Peilin Zhao, and Wei Chen. 2016. Repeat buyer prediction for e-commerce. In *KDD*. ACM.

T Mikolov and J Dean. 2013. Distributed representations of words and phrases and their compositionality. *NIPS*.

John M Prager. 2006. Open-domain question: answering. *Foundations and Trends in Information Retrieval*, 1(2):91–231.

Aliaksei Severyn and Alessandro Moschitti. 2013. Automatic feature engineering for answer selection and extraction.

Ellen M Voorhees et al. 1999. The trec-8 question answering track report. In *Trec*, volume 99, pages 77–82.

Mengqiu Wang and Christopher D Manning. 2010. Probabilistic tree-edit models with structured latent variables for textual entailment and question answering.

Di Wang and Eric Nyberg. 2015. A long short-term memory model for answer sentence selection in question answering.

Mengqiu Wang, Noah A Smith, and Teruko Mitamura. 2007. What is the jeopardy model? a quasi-synchronous grammar for qa.

Yi Yang, Wentau Yih, and Christopher Meek. 2015. Wikiqa: A challenge dataset for open-domain question answering.

Wen-tau Yih, Xiaodong He, and Christopher Meek. 2014. Semantic parsing for single-relation question answering. In *ACL*, pages 643–648. Citeseer.

Lei Yu, Karl Moritz Hermann, Phil Blunsom, and Stephen Pulman. 2014. Deep learning for answer sentence selection. *arXiv preprint arXiv:1412.1632*.

Yuanzhe Zhang, Shizhu He, Kang Liu, and Jun Zhao. 2016. Attention mechanism on question answering over knowledge bases. In *AAAI*.