BERT-BASED AND ATTENTION-BASED FRAMEWORK FOR COMMUNITY QUESTION ANSWERING TASKS

XUAN ZHAO

A THESIS
SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF APPLIED SCIENCE

GRADUATE PROGRAM IN
ELECTRICAL ENGINEERING AND COMPUTER SCIENCE
YORK UNIVERSITY
TORONTO, ONTARIO
MAY 2021

Abstract

Community question answering (CQA) becomes increasingly prevalent in recent years, providing platforms for users with various background to obtain information and share knowledge. However, there are a large number of answers, difficult for users to view item by item and select the most relevant one. Therefore, answer selection and duplicate question detection become very significant subtask of CQA.

In this work, we propose different models to explore these tasks. First, we study the correlation between question and paired answer. Then, we introduce the attention-based model Himu-QAAN for the answer selection task. Also, we present a BERT-based model Bert-QAnet for duplicate question detection task. We test our methods on various datasets. The results show that our methods achieve significant performance.

Acknowledgements

It has been a great journey studying under **Professor Jimmy Huang**'s supervision. He is the most dedicated researcher that I have ever known. I have to say that without his instruction, support, and inspiration, I could not finish my work during this pandemic that hit many peoples' life so hard. I would like to express my sincere gratitude to him.

The days working at the *Information Retrieval and Knowledge Management Research Lab* would definitely become one of the most valuable parts of my memory. I would like to thank all my fellow researchers working at this lab for their great cooperation.

I would also like to thank my thesis committee member, **Professor Mokhtar Aboelaze and Professor Xiaohui Yu** for their dedication, as well as for spending their valuable time on my thesis.

I would like to thank Dr. Haitian Yang. Cooperating with him has always been very pleasant and inspiring.

Last but not the least, I would like thank Vangi Munchinsky for her support. Her extraordinary personality shows the true beauty of Canada.

Table of Contents

List of Tables

List of Figures

Preface

This thesis is submitted to the Faculty of Graduate Studies in partial fulfillment of the requirements for a Master of Applied Science Degree in Computer Science.

Some work during the grauate studying has been accepted or under review for publication as:

(1) Xuan Zhao, Jimmy Huang, Haitian Yang. "CANs: Coupled-Attention Networks for Sarcasm Detection on Social Media." Proceedings of the IEEE International Joint Conference on Neural Network (IJCNN), 2021.

(1) Xuan Zhao, Jimmy Huang. "Bert-QAnet: BERT-encoded Hierarchical Question-Answer Cross-Attention Network for Duplicate Question Detection." Neurocomputing (under review).

1 Introduction

1.1 Motivation

As a powerful product of the digital revolution, Internet has revolutionized the way people obtain information, provide and share knowledge. One of the most common ways is to type keywords explicitly to express the requirements, while search engines usually return a large amount of web pages which vary in relevance to the submitted keywords. Users can browse one by one and select. However, conventional search engines can only provide a general solution to domain-specific problem, and in some cases, there is no guarantee of the desired answer. Therefore, in order to partly overcome the shortcomings, Community Question Answering (CQA) has become popular in recent years. With the advent of question answering communities during the past few years, such as Yahoo! Answers[1], Stack Overflow[2],

[1] 1https://answers.yahoo.com

[2] https://www.stackoverflow.com

Quora[3], Baidu Knows[4], etc., more and more users can search informations that they are interested in, post questions of their own concerns, provide and share knowledge. Most CQA systems allow users to ask questions without any subject restrictions (such as Yahoo! Answers, Quara, Baidu Knows), while the few CQA systems focus on specific areas (for example, only programming-related questions can be posted on Stack Overflow).

Although works as a web-based services, Q&A systems can provide an interactive experience, and more important, a faster retrieve in specific fields compared to conventional search engines. The main purpose of Q&A system is to provide various candidate answers to the posted questions and match the most relevant answers in the shortest time. The biggest difference between traditional information retrieval systems and CQA system can utilize tacit knowledge (here refers to valid information embedded in different communities) or explicit knowledge (here refers to effective information embedded in all archived question-answer pairs) to answer a lot of new questions posted daily.

In detail, the Q&A community processes as the followings: (1) Questioner types a question, and after the system confirms that the question has no inappropriate content, the question will be posted in the community and awaited answers. (2)

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

[4]https://zhidao.baidu.com/

Other users interact in two ways: post desirable (or indesirable) answers based on their opinions / interests / expertises; vote "like" or "dislike" to different answers based on the validity, importance, and content of the response. Finally, based on the questioner's satisfaction, he / she can mark an answer as the best answer (selected by the highest number of votes, or by the questioner himself), and then the question can be archived. Some scholars have named this process as "Question Lifecycle" and summarized it into the four steps: question creation; question answering; question closing; and question search [72, 102, 83]. In particular, question closing means that the questioner selects the most relevant answer that meets his needs as the best answer, or the system determines the best answer based on the number of votes if the answer fails to meet the questioner's intention. Both can terminate the process of question answering. Problem search is using archived question and answer pairs to solve new problems, which usually is realized by full-text search or navigation in taxonomy of question topics.

Research related to the Q&A community can be broadly divided into the following categories: System-wide analysis, usually covering sub-research topics such as community characteristics, knowledge sharing processes, design principles, transferability, and mobile use; content-related, which usually covers Q&A quality, friendliness of posts, innovation diffusion, personalized posts; user analysis, which typically include sub-topics like user context, expertise and reputation, question and answer

Question Subject	Checking the history of the car.
Question Body	How can one check the history of the car like maintenance, accident or service history. In every advertisement of the car, people used to write "Accident Free", but in most cases, car have at least one or two accident, which is not easily detectable through Car Inspection Company. Share your opinion in this regard.
Answer 1	Depends on the owner of the car.. if she/he reported the accident/s i believe u can check it to the traffic dept.. but some owners are not doing that especially if its only a small accident.. try ur luck and go to the traffic dept.
Answer 2	How about those who claim a low mileage by tampering with the car fuse box? In my sense if you're not able to detect traces of an accident then it is probably not worth mentioning... For best results buy a new car :)

Table 1.1: An example question and the selected candidate answers in CQA.

selection, abuse behavior, etc.

A typical CQA example is shown in Table 1. In this example, Answer 1 is a

good answer because Answer 1 gives "check it to the traffic dept", which is the most valuable piece of information and the most suitable for the intention of the questioner. Answer 2, although related to the problem, does not provide useful information, and therefore is considered a bad answer.

From the given example in Table 1, we can observe the characteristic of questions in CQA. It is also one of the most significant difference from conventional search. The description of a question usually consists of two parts, question subject and question body. Question subject is mainly a brief summary of the question, usually containing the key words of the question. Question body is a detailed description of the question, including critical information, as well as some extended information of the question. Question subject often is as simple as a title, while question body are longer in length than question-subject, and naturally more informative. Also, we observed that words and phrases of questions are very colloquial, but paired answers especially the best answers often are rich in vocabulary and highly specialized.

In general, users of CQA can be simply categorized into two types, expert users and common users. Expert users often specialize in a certain domain, who are able to provide very detailed and professional answers. These answers are more likely to demonstrate the essential intention of the question, and more informative than questions themselves. Common users are more likely to ask question and only provide simple answers to some posts. As a result, answers from different

users show huge variance in quality, some are only partially related to the topic, some even completely deviate from the actual intention of user, while some are fairly desirable. It is time-consuming for users to view them all and select the answers that might be the most relevant. Furthermore, with the rapid growth of such websites, the number of questions without having been answered increases significantly. The majority of these questions usually have been posted by other users and answered before, might not present with the same words and syntax, but mostly express the same semantics. Therefore, more and more new problems make CQA systems without proper collaboration support overweight due to user requests. This inevitably results in that users are not able to receive a desirable answer within an acceptable time, and hence the main goal of the CQA system cannot be achieved. To solve this issue, many techniques such as adaptive support approaches and question routing have been proposed. Adaptive support methods are based on the results of research on content and user modeling, and strive to directly impact users' collaboration.

For both question and answer, there are a vast amount of redundant information, these redundancies affect the performance of answer selection solutions. Bian et al.'s research [57] proves that the above observations are consistent with the common phenomenon of QA system. A study he conducted found that users are more likely to use CQA forums to seek opinions and answers to non-factoid questions than

factoid questions [4]. He attributes the success of the Q A community to the fact that they enable users to get accurate answers to natural language questions directly and efficiently from the community. For questions with too many answers, answer selection can significantly reduce users' waiting time, and provide better experience [13]. For questions without having been answered, answer selection can provide the best answer that has been posted before.

Hence, question classification and answer selection become very important subtasks of CQA [98, 39, 47, 73, 29]. The former aims to assign new posted question to a specific preset category, while the latter aims to find the most relevant answer in repository. Specifically, question classification task uses archived question-answer pairs to solve new-coming posts from users, to facilitate users to answer the questions that belong to their familiar areas more efficiently.

For answer selection task and duplicate detection task, often there are two application scenarios. The first application scenario is to use CQA information retrieval technology to identify whether a given query is semantically equivalent to an existing question in the repository. After retrieval, the paired answers of the existing question are referred to users as the relevant candidates. The second application scenario is that treating the existing question and answer in the archive as "question-answer pair" to determine whether the question-answer pair is the best match, which also can be explained as converting answer selection task to classi-

fication task. With this answer selection technique, questions and corresponding answers in CQA can be organized more effectively and efficiently. In this research, we focus on improving the performance of technique that is applied to the second application scenario.

1.2 Challenges

However, it is difficult to assign questions to pre-defined categories in the community since a large number of synonyms, semantic features, and syntactic features in natural language. For example, "What are good ways to look for good local restaurants?" and "How do you search for great restaurants along your route?" both are seeking good places to eat, but rarely contain almost identical vocabulary and syntactic features.

- Therefore, The conventional methods based on term frequency are not able to identify the difference of sentences with similar vocabulary and syntactic features.

- Also, due to the subjectivity of users, the redundancy is prevalent in CQA, which makes it even more difficult to identify the semantic meaning of different questions.

Answer selection in community is also challenging, even some work on this task

[89, 97, 76, 34] has shown the effectiveness of targeting the high-quality questions. The reasons can be described from two aspects.

- First, questions and answers often contain auxiliary verbs, which in the majority cases cannot provide useful information. A large number of synonyms, semantic features, and rich syntactic features make the selection task even trickier. Many researchers focus on techniques that treat each word in question and answer equally [76, 34]. But due to the redundancy and noise [104, 94], only part of the content in the answer is informative. Therefore, a vast amount of deviation cannot be avoided, which sometimes causes the selected answers are not suitable for users.

- Secondly, a posted question consists of two parts, question subject and question body. The effects of these two parts on selection performance need to be studied. Effectively use the relationship between question subject and question body to solve answer selection task is another challenge.

1.3 Proposed Framework

Many previous methods mainly focus on solving the first issue, and some researchers [79, 4, 100, 67, 52] extract lexical features or thread-level features to represent QA pair. But the difference between question subject and question body is ignored.

- In this research, first, a Question Answering Attention Network (QAAN) is proposed for question classification of CQA, which uses attention mechanism to assign different attention weights to questions and their paired answer. QAAN studies the paired answers in CQA to improve the performance of question classification task. Secondly. And, QAAN uses Attention mechanism to captures the attention weights both in question and paired answer, extracting more semantic features.

- Secondly, Hierarchical Multi-Layer Question Answer Attention Network (Himu-QAAN) is proposed, a hierarchical question-answer cross-attention model for answer selection in community question answering. This model uses cross attention mechanism between question and answer words to discern the most informative words that are essential for providing an adequate answer. Then Himu-QAAN use a hierarchical inner attention, firstly for different words in sentence, and then for different sentences in answer, to consecutively capture the answer features that are possibly most relevant to the question. In the last step, model Himu-QAAN compute the matching between each given question and candidate answer pair.

- Thirdly, a novel approach based on BERT (Bidirectional Encoder Representations from Transformers) for duplicate question detection, namely, the BERT-

encoded Hierarchical Question-Answer Cross-Attention Network (Bert-QAnet) is proposed, which incorporates answer information as an external resource to obtain more semantic features at word, sentence, and document level, while eliminating redundancies and noise. To integrate answer information and obtain textual relevance, we use three heterogeneous attention mechanisms, which are cross-attention, word inner attention, and sentence inner attention. Specifically, cross-attention is good at finding the correlation between question and answer, word inner attention concentrates on obtaining word-level semantic, and sentence inner attention captures sentence-level features. These features eventually are feed to the presentation layer as inputs.

1.4 Contributions

The proposed model uses the relationship between question-subject and question-body as important information for answer selection. First, cross attention mechanism between question subject and question body is used, as well as hierarchical cross attention mechanism between question subject and answer, respectively. Secondly, a hierarchical inner attention is used, for different words in a sentence, also for different sentences in an answer, to consecutively capture the answer features that are possibly most relevant to the question. Finally, the final results are obtained by integrating these mechanisms mentioned above.

The main contributions of this work can be summarized as follows:

- For investigating the role of answers'information corresponding to the questions, QAAN studies the correlation between question and paired answer, taking question as the primary part of the question representation, and the answer information is aggregated based on similarity and disparity with the answer.

- Himu-QAAN treats question-subject and question-body differently. Specifically, cross attention is used between question-body and answer, and moreover choose question-subject as benchmark, to effectively obtain important words in question-subject. In the Ablation Study, we implement the verification experiment. The results show that using attention mechanism can effectively solve the task of answer selection.

- BERT-QAnet uses BERT as a textual feature extraction approach to achieve better representations. And then, *paired answer* is added as an external resource for duplicate detection. To make full use of answer information, each answer is divided into various sentences of fixed length. Further, cross-attention is used between question and answer to acquire more accurate correlations.

1.5 Outlines

The remaining of this work is organized as follows: Section 2 introduces other researchers' work about question classification, answer selection, and duplicate detection; Section 3 describes the proposed models, QAAN, Himu-QAAN, and BERT-QAnet, explains various parameters and the training process; in Section 4, the implementations of QAAN, Himu-QAAN, and BERT-QAnet are discussed, also the proposed models are compared with other baselines; Section 5 summarizes the conclusion and suggests the further research potentials in the future.

2 Literature Review

In this chapter, the recent work of machine learning models and deep learning models in question answering are discussed. The literature review focuses on two tasks of question answering that this work aims to solve. These tasks are described specifically, i) Question Classification task, and ii) Answer Selection task. For each task, classic conventional models are listed in the literature review. Then, the attention-based architectures and state-of-the-art models apply attention mechanism that inspire this work are described.

2.1 Question Classification Task

Question classification is one of the tasks in natural language processing (NLP). Most of the conventional natural language processing tasks are based on statistical machine learning approaches. With the rapid development of deep learning [89], more and more researchers and scholars have applied deep learning to natural language processing tasks. Adhikari et al. [97] questioned the complexity of

the existing neural network architecture for document classification, and proposed embedding dropout, weight dropping, and temporal averaging in the training process of simple Bi-LSTM. This model has achieved good performance on different datasets. The recent research by Melis et al. [76] showed that the fine-tuned model based on the standard LSTM outperform other models. Vaswani et al. [34] showed that the model uses attention mechanisms respectively can achieve the comparable performance with the model uses an encoder with attention mechanism to convert sequences. This model has demonstrated that most of the complex neural network mechanisms are not imperative. Mohammed et al. [43, 42, 41] illustrated that Vanilla RNN and basic CNN models can achieve better results in knowledge-based questions and answers than complex architecture neural network models. Sculley et al. [1] claimed that the lack of rigor in domain knowledge can be easily solved by removing the noise. Lipton et al. [48] also agreed with these observations, clarifying that many authors often use fancy data formulas to confuse or impress reviewers rather than clarify factual issues. Yang et al. [77] proposed a sequence generation model (SGMs) based on encoder-decoders to generate a pair of labels for each document. This model has achieved good results on the relevant data sets. These are some recent models that have performed well in natural language processing tasks.

In recent years, with the rapid development of deep learning, which has been widely used in the domain of natural language processing. As the focus of many

NLP researchers, the Community Question Answering has been flourished with a number of state-of-the-art models and algorithm. Kim et al. [51] studied the deep learning model - convolutional neural network (CNN) with trained word embedding for sentence classification task. Kalchbrenner et al. [65] developed a dynamic convolutional neural network (DCNN) that learns sentence semantics by simulating semantic information through the DCNN network for question classification. DCNN used a global k-max pooling operation to solve the issue that the sentences with different lengths, also learned the dependence of lengths for different sentences. Le et al. [46] develpled a forest convolutional neural network (FCN) using forset as input of convolutional neural networks. Random increase or decrease of branches was realized in FCN. Experiment results demonstrated that FCN has achieved state-of-the-art results in both sentiment analysis and question classification tasks. Mou et al. [95] proposed a tree-based convolutional neural network (TBCNN) with the sentence-dependent syntactic tree and component tree. TBCNN used the extracted the structural features of sentences as components, applying the maximum pooling to merge multiple features. Komninos et al. [36] studied the effects of word embeddings on deep neural networks. The results showed that context-based word embedding achieved better performance in sentence classification tasks.

2.2 Answer Selection Tasks

Since answer selection became a very important subtask for community QA, researchers have been racking their brains to discover new features, design new models for effective solutions. QA community contains a large number of interactive information from real users in different fields, which is an important knowledge source for enhancing the repository of automatic QA system. Many of today's popular QA communities use the constantly updated resources to update and augment the system's knowledge base, such as Siri3 and Watson4. Answer selection task is to study how to evaluate each candidate answer based on a given question, and then select the answer that matches the user's question. Based on the majority of existing research in the past, the main difficulty lies in how to effectively establish the semantic relationship between question and answer. The existing answer selection techniques can be roughly divided into four categories: feature engineering based methods, syntax tree based methods, translation models, and deep learning models.

2.2.1 Conventional Methods

In the early years, the answer selection task highly relied on feature engineering, linguistics tools and other external resource. Nakov et al. [55] studied a wide range

of feature types, including similarity feature, content feature, meta-feature, and feature automatically extracted from the SemEval CQA model. Tran et al. [76] research used the topic model- based features and word embedding-based features for answer ranking task. Filice et al. [19] designed various heuristic features and thread based-features, which can provide better selections. Even these techniques achieved good performance, the highly dependence on feature engineering result in the indispensability of domain knowledges and an enormous amount of handcrafts.

Also, to successfully accomplish the answer selection task, both semantic and syntactic features are necessary. Following the idea that by studying the syntactic matching between questions and answers, the most relevant candidate can be obtained by loose syntactic alteration. Wang et al. [82] designed a generative model based on the soft alignment of quasi-synchronous grammar by matching the dependency trees of question answer pairs. Heilman et al. [23] used a tree kernel as a heuristic algorithm to search for the smallest edit sequence between parse trees. Then the features extracted from these sequences are passed through a logistic regression classifier to compute the probability that whether an answer is relevant to the given question.

Also, Shtok et al. [70] used statistical methods to compute the probability that a satisfactory answer from the repository of solved question-answer pairs can be provided for a specific new question. This model successfully reused the past

answers of high quality and achieved good performance on an offline data set. Riahi et al. [62] used two statistical topic models, the Segmented Topic Model and the Latent Dirichlet Allocation model, respectively, to find experts in communities, and achieved good performance on a dataset extracted from the Stack Overflow platform. Also, the ablation experiment showed that compared to the LDA model, the Segmented Topic Model achieved even better improvement. Chen et al. [8] applied machine learning technique to build a predictive model based on text and metadata features. Their model predicted users' intention into three categories and recommended relevant answers based on the classification. It is not difficult to see from the above description that even some other research topics are inseparable from the support of the question retrieval task, thus, to some extent, question retrieval task is the basis of studying CQA.

Moreover, in the early years, many researchers focused on frequently asked questions (FAQ) for question retrieval and various methods have been proposed. Burke et al. [6] designed a FAQ system FINDER which used frequently occurred files as knowledge base, combining word similarity and semantic similarity for question retrieval. Berger et al. [5] solved the word sparsity problem among questions by learning a variety of statistical models to improve the performance of answer-finding. Jijkoun et at. [32] applied unsupervised learning to extract question-answer pairs from FQA web pages, and then modeling the question retrieval with vector

space model. Riezle et al. [63] proposed a statistical machine translation model for query expansion, aiming to narrow the lexical gap between query words and answer words by assigning synonyms using sentence paraphraser. Most of the techniques above focused on solving the words mismatch problem between query and answer, however, only the superficial meanings of words were captured, the information of topic categories hidden in the sentences were ignored.

Some early researchers studied various rules and statistical features, and then fed these features into machine learning models. Roth [64] proposed a sparse network for multi-class classification, which showed a good classification performance on high-dimensional data. This model learns a linear function for each category. The probability of the question in each category is calculated by this function. The methods commonly used for updating rules are naive Bayesian, perceptron, etc. Metzler et al. [49] applied radial basis kernel function in SVM for factoid questions, which used multiple feature fusion method to combine both syntactic and semantic features.

Another scenario of answer selection is to use information retrieval technology to determine whether a query is semantically equivalent to an existing question in the community [2, 103, 106]. Specifically, [103] calculates four similarity scores ,based on this using continuous word vectors of deep learning, topic model features, and phrase pairs in machine translation system to mine frequently duplicate ques-

tions that occur simultaneously. Hoogeveen et al. [24] observe that for detection of wrong labels, metadata features can capture user authoritative information, question quality, and the relationship between questions more powerfully, compared to plain text. Wu, Yan, et al. [91] used distributed index and MapReduce framework to calculate the similarity of question-answer pair, and then efficiently identify redundant data in a scalable way. However, distributed representation is an effective technique to solve the lexical semantic gap. Researchers have designed various similar features based on word embeddings [21], or demonstrated questions via neural networks to calculate similarity [16]. Also, some researchers proposed method [105] combining neural networks with FrameNet to achieve question matching.

Since the essence of answer selection is a ranking task, some previous studies propose to use the local ranking function for global ranking strategies. Jeon et al. [30, 31] compared various retrieval model, such as vector space model, language model, and translation model, in terms of query performance in question answering community. To the best of our knowledge, Barron-Cedeno et al. [3] is the first to use the structured prediction model for answer selection task. Joty et al. [14] used global inference process to investigate all the answers the answer-thread and represents them as fully connected graph.

Although these methods have shown good performance, it is difficult to represent structured features and solve data sparsity problem because only feature

vectors are taken as objects. Collins et al. [11] proposed a tree kernel method to compare the similarity between syntactic trees by calculating the number of identical tree segments, but the depth features and syntactic features of the nodes were disregarded. Wang et al. [81] used the tree kernel to model the structural features. The similarity between two sentences were calculated, but the semantic information was neglected.

Some researchers focus on studying different translation model, like word-based translation model, or phrase-based translation model for subtask of CQA. Murdock et al. [54] proposed a simple translation model for sentence retrieval in factoid question answering. Zhou et al. [107] proposed a phrase-based translation model, which aims to find semantically equivalent questions for new queries from the Q & A archives. Singh [71] extended the translation model using semantic entities to retrieve vocabulary and semantically similar issues, and used neighborhood-based classifiers to classify new issues. Wu et al. [88] designed an intent-based model by combining a translation-based model with a user intent classification. L. Chen et al. [7, 8] proposed a hybrid approach that combines classic query-likelihood language model and translation based language model to form an intent-based language model. Jeon et al., their subsequent work [93] of [30, 31] proposed a translation-based language model, which integrated the corresponding answers to the questions as supplementary.

2.2.2 Deep Learning Models

With the rapid development of deep learning [41], more and more researchers and scholars have applied deep learning to natural language processing (NLP) tasks. Adhikari et al. [1] questioned the complexity of the existing neural network architecture for document classification, and proposed embedding dropout, weight dropping, and temporal averaging in the training process of simple Bi-LSTM. This model has achieved good performance on different datasets. The recent research by Melis et al. [48] showed that the fine-tuned model based on the standard LSTM outperform other models. Vaswani et al. [77] showed that the model uses attention mechanisms respectively can achieve the comparable performance with the model uses an encoder with attention mechanism to convert sequences. This model has demonstrated that most of the complex neural network mechanisms are not imperative. Mohammed et al. [51] illustrated that Vanilla RNN and basic CNN models can achieve better results in knowledge-based questions and answers than complex architecture neural network models. Sculley et al. [65] claimed that the lack of rigor in domain knowledge can be easily solved by removing the noise, which has been by many examples in the paper. Lipton et al. [46] also agreed with these observations, clarifying that many authors often use fancy data formulas to confuse or impress reviewers rather than clarify factual issues. Yang et al. [95] proposed a

sequence generation model (SGMs) based on encoder-decoders to generate a pair of labels for each document. This model has achieved good results on the relevant data sets. These are some recent models that have performed well in natural language processing tasks.

In recent years, with the rapid development of deep learning, which has been widely used in the domain of natural language processing. As the focus of many NLP researchers, the Community Question Answering has been flourished with a number of state-of-the-art models and algorithm. Kim et al. [36] studied the deep learning model - convolutional neural network (CNN) with trained word embedding for sentence classification task. Kalchbrenner et al. [35] developed a dynamic convolutional neural network (DCNN) that learns sentence semantics by simulating semantic information through the DCNN network for question classification. DCNN used a global k-max pooling operation to solve the issue that the sentences with different lengths, also learned the dependence of lengths for different sentences. Le et al. [40] develpled a forest convolutional neural network (FCN) using forset as input of convolutional neural networks. Random increase or decrease of branches was realized in FCN. Experimental results demonstrated that FCN has achieved state-of-the-art results in both sentiment analysis and question classification tasks. Mou et al. [53] proposed a tree-based convolutional neural network (TBCNN) with the sentence-dependent syntactic tree and component tree. TBCNN used the ex-

tracted the structural features of sentences as components, applying the maximum pooling to merge multiple features. Komninos et al. [38] studied the effects of word embeddings on deep neural networks. The results showed that context-based word embedding achieved better performance in sentence classification tasks.

Compared to other traditional machine learning techniques, deep learning model does not require complex feature engineering, does not rely on large-scale external resources or annotation tools. Deep learning model can effectively capture the structured semantic features hidden in language expression through a multi-layer network, thereby provide better connection to lessen the semantic gap between question and answer.

Most of the deep question answer matching models use Deep Belief Networks (DBN) [27, 80] and Convolutional Neural Networks (CNN) [101, 61]. Especially, CNN has an even stronger ability for representing the structural features of sentences, which uses convolutional operations to encode consecutive words in sentences, is widely used in sentence modeling [36], sentence matching [26], and Q&A pair matching.

In CQA, as well as query answer pair, the context of answer is another important factor in determining the quality of the answer. Zhou et al. [109] proposed RCNN model, which uses the deep recurrent neural network to learn the content association of answers. RCNN is one of the few studies that takes the context factor into

consideration. Wan et al. [78] proposed MV-LSTM based on bi-direction LSTM to represent questions and answers. These representations are then passed through a tensor layer at each time step to capture the positional information. Zhang et al. [104] introduced a novel cross-attention mechanism to overcome the redundancy and noise which usually are prevalent in CQA. Shen et al. [69] used translation matrix to learn word representations, then calculated the relevance score between question and answer for each pair in the repository. Tay et al. [74] propose temporal gates for jointly learning the representations of sequence pairs, which ensure that question-answer pairs are aware of what each other is forgetting or remembering. Even this model is simple and effective, the relationship between question subject and question body is ignored which is very helpful for condensing the task. Wu et al. [89] proposed a question condensing networks that uses the question subject and question body relationship to align question-answer pair. Zhou et al. [108] introduced a recurrent convolutional neural network which combines the characteristics of the two structures to explore the semantic matching between query and answer, to capture the semantic correlations concealed in the word sequence of answers.

2.3 Duplicate Question Detection

There are a large number of studies on the duplicate detection task in CQA. Here we mainly review works that used deep learning technique, which is the most related

to our research.

Generally, two main challenges of duplication detection are lexical gaps and basic component matching. Distributed representations are effective in resolving the problem of lexical gaps. For the second challenge, researchers have either designed various similar features based on word embeddings [10, 16, 28], or analyzed questions using neural networks to calculate similarities [105, 44].

The following are some classic examples of solutions for detecting duplicate questions. [21] used distributed-index and MapReduce framework to calculate similarities of Q-A pairs, and then pointed out redundancies in a scalable way. [50] detected duplicate questions in Stack Overflow by calculating similarity scores of titles, descriptions, potential subjects, and labels of each Q-A pair. [106] used deep learning based continuous word vectors, topic model features, and phrase pairs of machine translation systems to mine duplicate questions. [91] observed that, compared to plain text, metadata features can capture users' authoritative information, question quality, and the relationship between questions more effectively.

3 Question Classification Task

In this chapter, the proposed hierarchical question-answer cross-attention model (QAAN) model is presented, which use the relationship between these question subject and question body as important information for answer selection. First, the task in mathematical form is introduced. Second, the framework of each proposed models is described. Furthermore, the structure and operations of the proposed models over each layer are detailed.

3.1 Question Classification Task Description

With the characteristics of CQA, the question-answer pair and classification results are described as a tuple of three elements (Q,A,y), where Q=$[q_1, q_2, \ldots, q_n]$ represents a question whose length is N. Each q_i is encoded by a one-hot vector, whose dimension is the same as the dimensions of vocabulary M. $y \in Y$ indicates the category corresponding to question. Therefore, the task is defined as, given a question-answer pair {Q, A}, the distribution of probability $Pr(y|Q, A)$ is modeled

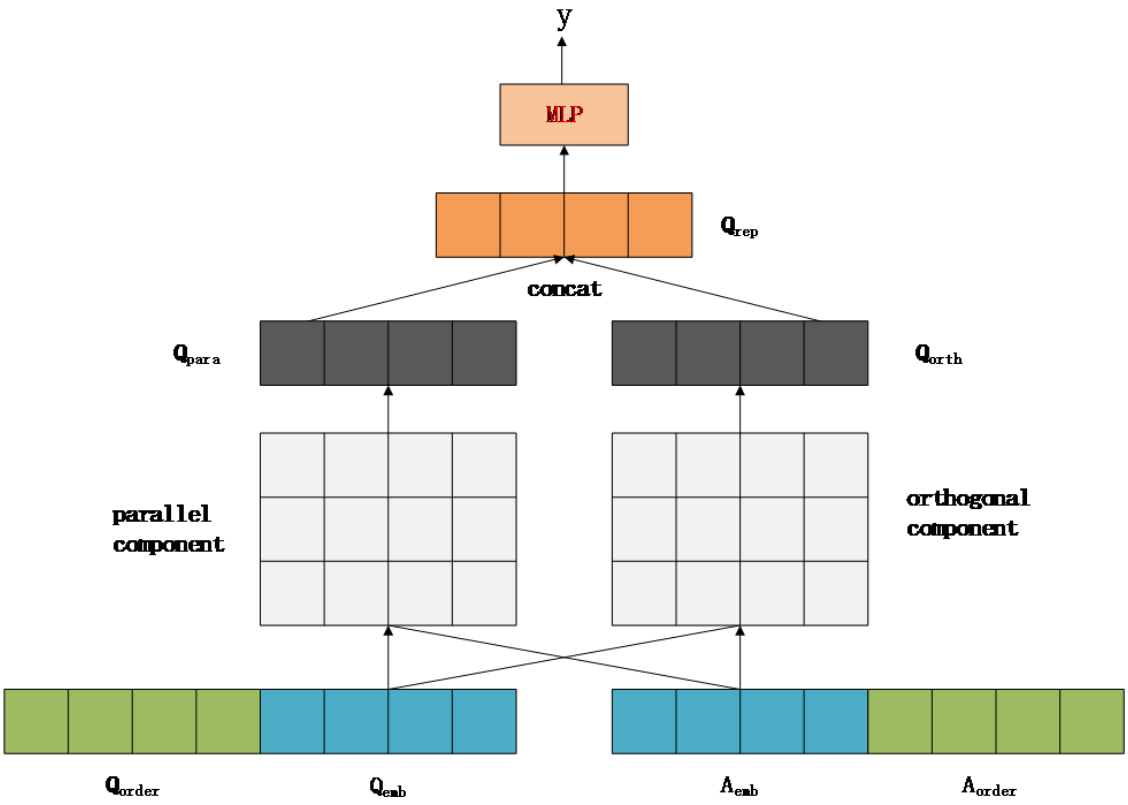

Figure 3.1: Overview of our proposed QAAN model.

by QAAN. The label of the category with the maximum probability distribution is assigned.

3.2 Overview of the Proposed Model QAAN

The corpus used in this research is extracted from the Yahoo!Answers website. For each question we ensure that there is at least one paired answer. Each question is labeled as one of the categories. Two set of word embeddings are obtained on

the corpus by implementing the character embeddings [37] and GloVe [59], respectively. The two embeddings are concatenated to preserve the validity of the position information effectively. The concatenation is used as the input the embeddings.

The pipeline of our QAAN model is illustrated as Fig.1.

Q_order represents the position information for each word, *Q_emb* is the concatenation of the two question embeddings. Similarly, *A_emb* is the concatenated word vectors in the corresponding answer, *A_order* indicates the position information of each word in the corresponding answer.

3.3 Word-Level Embedding

The word-level embeddings are composed of two modules: the Glove model proposed by Pennington et al. [59] which is trained on the Yahoo!Answers corpus, and character embedding proposed by Kim et al. [37]. The concatenation of the two embeddings provide various advantages. The relationship of words is captured more accurately and precisely when the embedding is trained on the extracted corpus since the texts in CQA are different in grammar and spelling from News and reports. Also it has been proved that character embedding is effective for OOV (out-of-vocabulary), especially for CQA tasks.

The two embedding vectors are concatenated to form a word-level embedding. For each question and the corresponding answer, the word embeddings are repre-

sented respectively as, $Q\_emb \in R^{d \times l}$ and $A\_emb \in R^{d \times m}$, d denotes the dimension of the concatenated word embedding.

3.4 Question Answer Attention Network

The question-answers are analyzed from the perspective of similarity and disparity to aggregate the information for better represent the question-answer relationship. We apply the orthogonal decomposition strategy proposed by Wang et al. [85] to achieve the representation. We embed the words of corresponding answers into two directions, horizontal and vertical. The formula is shown as follows (Equation (3.1)(3.2)):

$$a^{ij}_{para} = \frac{a^{j}_{emb} \cdot q^{i}_{emb}}{q^{i}_{emb} \cdot q^{i}_{emb}} q^{i}_{emb} \tag{3.1}$$

$$a^{ij}_{orth} = a^{j}_{emb} - a^{i,j}_{emb} \tag{3.2}$$

The length of vectors in the formula is d. The details of obtaining horizontal representation of paired answer is described, similarly, the process can be applied to the vertical direction. Q_{para} and Q_{para} are obtained during the process.

Q_{para} and Q_{orth} are passed through fully connected neural network to obtain multi-dimensional attention weights. Tanh is used as activation function, which

is similar to the method proposed by Shen et al. [68] To maintain a sufficient amount of output while preventing huge fluctuations in the final score, the following formula is used to normalize the output. The output for each dimension is shown as Equation (3.3):

$$b_{para}^{i,j} = c \cdot tanh(\frac{[W_{p1} a_{para}^{i,j} + b_{p1}]}{c}) \tag{3.3}$$

where $W_{p1} \in R^{d \times d}$ and $b_{p1} \in R^{d \times d}$ are parameters learned by QAAN, C is a manually tuned hyperparameter.

The word-level vector B is aligned by Equation (3.3). Then we normalize and expand the third party of Vector B to get the attention weight of each word in questions. The output is the weighted sum of the embedding of each word in the question divided by the embedding of the questions. The formulas are shown as: (Equation (3.4)(3.5)):

$$w_{para}^{i,j} = \frac{exp(b_{para}^{i,j})}{\sum_{j=1}^{m} exp(b_{para}^{i,j})} \tag{3.4}$$

$$q_{ap}^{i} = \sum_{j}^{m} w_{para}^{i,j} \odot a_{emb}^{j} \tag{3.5}$$

Where $\odot$ represents the point-wise product. The advantage of the multi-dimensional attention mechanism is that an optional feature for each word is extracted given

the context. We apply the fusion gate to unify the relationship between the words of questions and words of corresponding answers. The formulas are shown as: (Equation (3.6) (3.7)):

$$FGate_{para} = \sigma(W_{p2}Q_{emb} + W_{p3}Q_{ap} + b_{p2}) \tag{3.6}$$

$$Q_{para} = FGate_{para} \odot Q_{emb} + (1 - FGate_{para}) \odot Q_{ap} \tag{3.7}$$

where $W_{p1}, W_{p2} \in R^{d \times d}$ and $b_{p2} \in R^{d}$ are learned by the fusion gate. $FGate_{para}$, Q_{emb}, Q_{ap}, $Q_{para} \in R^{d \times l}$ and $Q_{rep} \in R^{2d \times l}$ are the representations of the questions. Q_{para} and Q_{orth} are concatenated representations. The question representation $Q_{rep} \in R^{2d \times l}$is passed through a two-layer feedforward neural network. The last layer is a softmax function which calculate the distribution probability $Pr(y \mid Q, A)$.

3.5 Dataset

The dataset in this paper is extracted from Yahoo!Answers. Each category is sorted based on the number of questions. 60 categories are selected with the largest number of questions, and the number of questions in these categories is more than 1000. All the samples are question-answer pairs to ensure that each question has a best answer. Therefore, 1000 question-answer pairs are randomly selected from

Number of Questions	200,998
Number of Answers	1,848,441
Number of Best Answers	201,075
Number of Classes	60
Number of Answers per Question	9.405

Table 3.1: Statistics of the original Corpus.

Number of Questions	60,000
Number of Answers	60,000
Number of Classes	60

Table 3.2: Statistics of the extracted corpus.

60 categories. 70% of sample is the training set, 20% is testing set, and 10% for validation. The statistics of the corpus is shown in Table (5.1-5.3):

3.6 Training and hyperparameters

The NLTK toolkit is used in the text preprocessing procedure for each question and corresponding answer, including capitalization conversion, stemming, removal of stop words, et al. The preprocessed dataset is trained to obtain 300-dimensioanl

	Training Set	**Testing Set**	**Validation Set**
Number of questions	42,000	12,000	6,000
Number of answers	42,000	12,000	6,000
Average length of question	10.26	10.08	11.26
Average length of answer	42.38	41.26	40.68

Table 3.3: Statistics of the extracted corpus.

initialized word vectors (GloVe). The vectors for out-of-vocabulary are set to zero. The algorithm we choose for optimization is Adam Optimizer, with momentum coefficient 0.9, the second momentum coefficient 0.999. The model is learned with initial learning rate $[1 \times 10^{-9}, 4 \times 10^{-5}, 1 \times 10^{-7}]$, L2 regularization parameters $[1 \times 10^{-6}, 4 \times 10^{-7}, 1 \times 10^{-7}]$, and batch-size $[64, 128, 256]$. We select the parameters with the best performance on validation set and then evaluate the final performance on the testing set.

3.7 Evaluation Metrics

In research communities, it is often necessary to use shared and comparable performance indicators to evaluate performances. Some examples of these metrics include recall rate, precision, accuracy, F metric, micro-average and macro-average. These

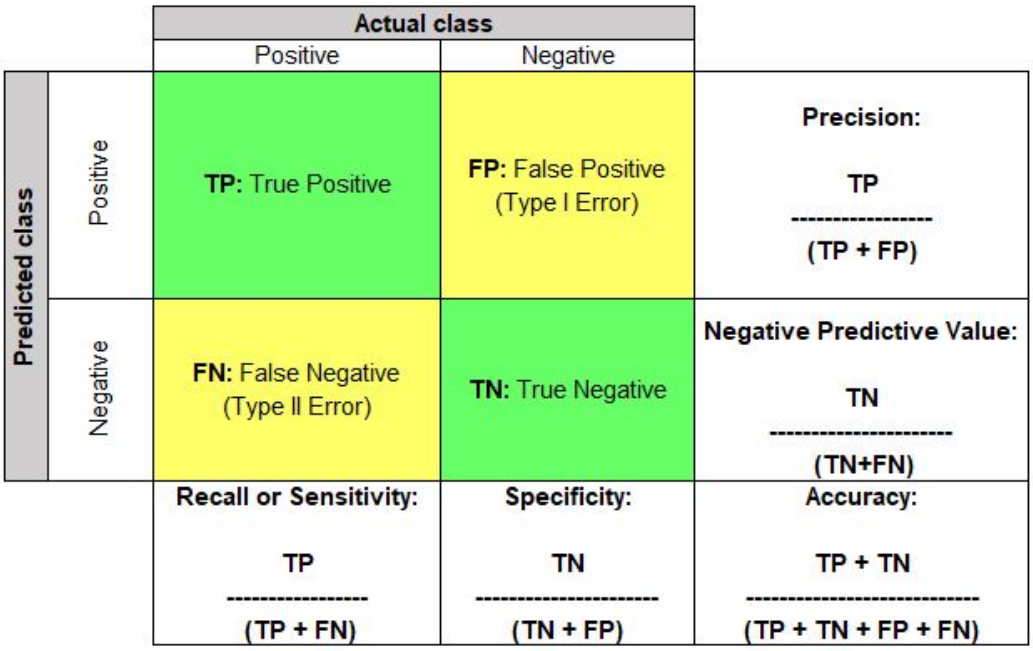

Figure 3.2: Performance Evaluation Measures.

metrics are based on the "confusion matrix" (as shown in Figure 3.2), which includes true positive (TP), false positive (FP), false negative (FN), and true negative (TN) .

As shown in Fig 3.2, the precision, recall, F1 score obtained by the dimensionality reduction algorithms using traditional feature extraction modules, basing on the text frequency.

Precision (Eq. 3.8) is the ratio of all correctly retrieved results (*True Positive*) to all the retrieved results (True Positive and False Positive). Therefore,

$$Precison = \frac{TP}{TP + FP} \tag{3.8}$$

Recall (Eqa. 3.9) calculates the proportion of all correctly retrieved results (*True Positive*) in all the results that should be retrieved (*True Positive* and *False Negative*), which is denoted as:

$$Recall = \frac{TP}{TP + FN} \tag{3.9}$$

As another evaluation index, F1 score (Eqa. 3.10) is the harmonic mean of *Precision* and *Recall*, is stated as:

$$\frac{2}{F_1} = \frac{1}{Precision} + \frac{1}{Recall} \tag{3.10}$$

which is used to combine the results of *Precision* and *Recall*, can also be demonstrated as (Eqa. 3.11):

$$F_1 = \frac{2PR}{P + R} = \frac{2TP}{2TP + FP + FN} \tag{3.11}$$

where, P stands for *Precision*, R stands for *Recall*.

The most ideal situation is definitely higher accuracy and recall rate. However, when the recall rate is increased, it sometimes affects the Precision, so the Precision can be regarded as a function of the recall rate, namely: $P = f(R)$, that is, with

the recall rate from 0 to 1, the change of Precision. Then the function $P = f(R)$ can be integrated on R, and the expected mean value of PP can be obtained. The formula is as follows (Eqa. 3.12):

$$AveP = \int_0^1 P(r)\mathrm{d}r = \sum_{k=1}^{n} P(k)\Delta(k) = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{N} \tag{3.12}$$

Where $rel(k)$ indicates whether the r$^{\text{th}}$ document is relevant. The value is 1 if relevant, otherwise 0. $P(k)$ represents the accuracy rate of the first k documents, N is number of relevant documents.

The calculation method of $AveP$ can be simply demonstrated as (Eqa. 3.13):

$$AveP = \frac{1}{R} \times \sum_{r=1}^{R} \frac{r}{position(r)} \tag{3.13}$$

Where R represents the total number of related documents. $position(r)$ demonstrate: the result list is viewed from the back, the position of the r-th related document in the list.

To average the $AveP$ of multiple query statements, that is, the mean of average precision scores, the formula is (3.14):

$$MAP = \frac{\sum_{q=1}^{Q} AveP(q)}{Q} \tag{3.14}$$

In this research, *F1*, *accuracy* (Acc) and MAP (Mean Average of Precision) are

used as the evaluation metrics to evaluate the performance of the QAAN.

3.8 Baseline Models

Five classification models are used as comparison models. As shown in Table 3.4, (1), (2), (3), (4), (5) are the BaseLine of QAAN. The detailed description of the these models are on page xx.

3.9 Results and Analysis

The experiment results in Table 3.5 demonstrate that:

1. JAIST and HITTZ-ICRC are two models with good performance on the data set SemEval2015. The experiment results show that JAIST outperforms BGMN model by 0.0095 in terms of MAP, 0.0119 in terms of F1, and 0.0098 in terms of Acc. In general, JAIST shows better performance than BGMN on three evaluation metrics, which proves that not all deep learning models can outperform conventional machine learning models. (Row 1 VS Row 3)

2. The ECUN model combining convolutional neural network with supervised learning outperforms BGMN, HITZZ-ICRC, and JAIST in terms of MAP, F1, and Acc. The results demonstrate that the combination of conventional ma-

Model	Reference and Description
JAIST	Proposed by [76]. It used an SVM classifier to incorporate various kinds of features, including topic model based features and word vector representations.
HITSZ-ICRC	Proposed by [25]. It combined ensemble learning and hierarchical classification method to classify answers.
BGMN	Proposed by [90]. It used memory mechanism to iteratively aggregate more relevant information which is useful to identify the relationship between question and answer.
ECUN	Proposed by [87] It combined a supervised model using traditional features and a convolutional neural network to represent the question-answer pair.
HAN	Proposed [58] It propose multilingual hierarchical attention networks for learning document structures, In our Experimental for question classification.
QAAN	Our model

Table 3.4: Baseline Models and Description.

Model	MAP	F1	Acc
JAIST	0.7473	0.6587	0.6635
HITSZ-ICRC	0.7304	0.6368	0.6256
BGMN	0.7378	0.6468	0.6537
ECUN	0.7658	0.6875	0.6938
HAN	0.7718	0.6928	0.7046
QAAN	**0.7784**	**0.7038**	**0.7126**

Table 3.5: MAP, F1 and Acc Performance of the Six Models on Relevant Corpus.

chine learning model and deep learning model can achieve better performance inn question classification task. (Row 4 VS Row 3, Row 2, Row 1)

3. The HAN (Hierarchical Attention) model outperforms ECUN, BGMN, FC-CRF, HITSZ-ICRC, and JAIST on three different evaluation indexes, MAP, F1, and Acc, respectively. The results demonstrate that a series of models based on attention mechanism can achieve better performance than conventional machine learning models and convolutional neural network models on question classification task, at least in the case of this paper. (Row 5 VS Row 4, Row 3, Row 2, Row 1)

4. Our QAAN model employs attention mechanism combining the answer infor-

mation with corresponding answer information. The experiment results show that QAAN outperform other five models in terms of three different evaluation metrics, which prove that corresponding answers can provide important information for question classification. QAAN can successfully learn these crucial resources. (Row 6 VS Row 5, Row 4, Row 3, Row 2, Row 1)

3.10 Ablation Study

To prove the validity of QAAN, in addition to the six baseline models above, six comparative experiments were conducted to demonstrate the improvement of QAAN. As shown in Tab 5.6.

The experiment results show in Table 5.7 show that:

1. Comparing Row1 VS Row2, it can be observed that the task-specific word embedding trained on the corpus show better performance, indicating that the questions in the community are distinct from text in News and web pages. This is because there are typos, syntactic errors, abbreviations due to the lack of professional background, which is a challenge for question classification.

2. Comparing Row7 VS Row1 Row2, QAAN shows better performances both with special-trained word embedding and character embedding, which prove that QAAN can also solve the out-of-vocabulary issue.

Methodology	Description
(1) without task-specific word embeddings	where word embeddings are initialized with the 300-dimensional GloVe word vectors trained on Wikipedia 2014 and Gigaword 5
(2) without character embeddings	where wordlevel embeddings are only composed of 600-dimensional GloVe word vectors trained on the domain-specific unannotated corpus
(3) question only	where only question is used as question representation
(4) answer only	where only answer body is used as question representation
(5) similarity only	where the parallel component alone is used in question - answer interaction
(6) disparity only	where the orthogonal component alone is used in question - answer interaction

Table 3.6: Experimental Methods and Description.

Model	**MAP**	**F1**	**Acc**
(1) without task-specific word embeddings	0.7236	0.6357	0.6838
(2) without character embeddings	0.7286	0.6374	0.6826
(3) question only	0.6756	0.6175	0.6365
(4) answer only	0.6852	0.6248	0.6475
(5) similarity only	0.7648	0.6985	0.7068
(6) disparity only	0.7538	0.6849	0.6988
(7) QAAN(our)	0.7784	0.7038	0.7126

Table 3.7: MAP, F1 and Acc of the Six Models on Relevant Corpus.

3. Comparing Row3 VS Row 4, paired answers provide more useful information than questions themselves, and the meanings of questions are represented better by the paired answers. This is mainly because, in question answering community, most of the questions is generally ambiguity. But the respondents, also known as, the providers of answers, have strong domain expertise. The answers are often more informative and represent the question better.

4. From the comparison of Row5 VS Row6, we can see that the parallel com-

ponents show better performance than vertical components in question classification task. This is because the parallel component can more comprehensively use the sequence information of texts to improve the classification performance.

4 Answer Selection Task

In this chapter, a hierarchical question-answer cross-attention model, the Hierarchical Multi-Layer Question Answer Attention Network (Himu-QAAN) is proposed, which is used for answer selection of community question answering. First, the task in mathematical form is introduced. Second, the framework of each proposed models is described. Furthermore, the structure and operations of the proposed models are detailed layer by layer.

4.1 Answer selection task description

In this research, the answer selection task of community question answering can be described as a tuple of four elements (S, B, A, y). $S = [s^1, s^2, \ldots, s^g]$ represents the question subject whose length is g. $B = [b^1, b^2, \ldots, b^m]$ is the question body whose length is m. $A = [a^1, a^2, \ldots, a^n]$ denotes the answer corresponding to the question whose length is n. $y \in Y$ represents the relevance degree that whether a answer can answer the question properly or not. More detailed,

$Y = \{Good, PotentiallyUseful, Bad\}$, where Good represents that the answer can provide a proper solution for the question, $\{PotentiallyUseful\}$indicates that the answer might provide the useful solution to the user, bad means the answer is not relevant to the question or useless to users. Generally, the answer selection task of Community Question Answering can be summarized as, given a set of $\{S, B, A\}$, our model Himu-QAAN canculates the conditional probability $Pr(y \mid S, B, A)$ and then assign a label with the maximum probability distribution to each answer.

4.2 Overview of the Proposed Model Himu-QAAN

The structure of Himu-QAAN can be described with six layers, including Embedding layer, Encoder layer, Word Inter Attention layer, Cross Attention layer, Adaptive Co-Attention layer, and Classification layer. These layers are distributed layer by layer from the bottom to the top. The pipeline of the proposed framework is demonstrated in Fig.2. The proposed model uses cross attention mechanism between question subject and question body, also hierarchical cross attention mechanism between question subject and answer, respectively, and then integrate them to get the final results. The model applies deep attention mechanism at word, sentence, document level, respectively, for selecting both factoid and non-factoid questions of various length.

In the following, the principle and role of each layer are introduced in detail

in accordance with the characteristics of the answer selection task in Community Question Answering.

4.3 Word-Level Embedding

Word embedding is one of the most classic techniques for vectorizing natural language. Among them, the two most widely used, most effective and most representative configurations are word2vec [50] and Glove [59]. Word2vec and glove are both unsupervised learning algorithms used to obtain word vector representations, but they differ in specific details.

Word2vec is developed by a group of Google researchers led by Tomas Mikolov, comprising two model architectures for displaying a distributed representation of words, they are respectively, continuous bag-of-words (CBOW) and skip-gram [50]. The CBOW model functions as predicting the focus word by the surrounding context. The skip-gram representation model predicts the surrounding context by using the focus word, which has been widely used and been proved to be particularly precise.

Glove was proposed by the Stanford NLP team, which trains aggregated global word-word co-occurrence statistics from the corpus, and the results show a linear substructure of the word vector space [59]. Manning Christopher D. summarizes the essence of GloVe as a log-bilinear model with weighted least squares as the

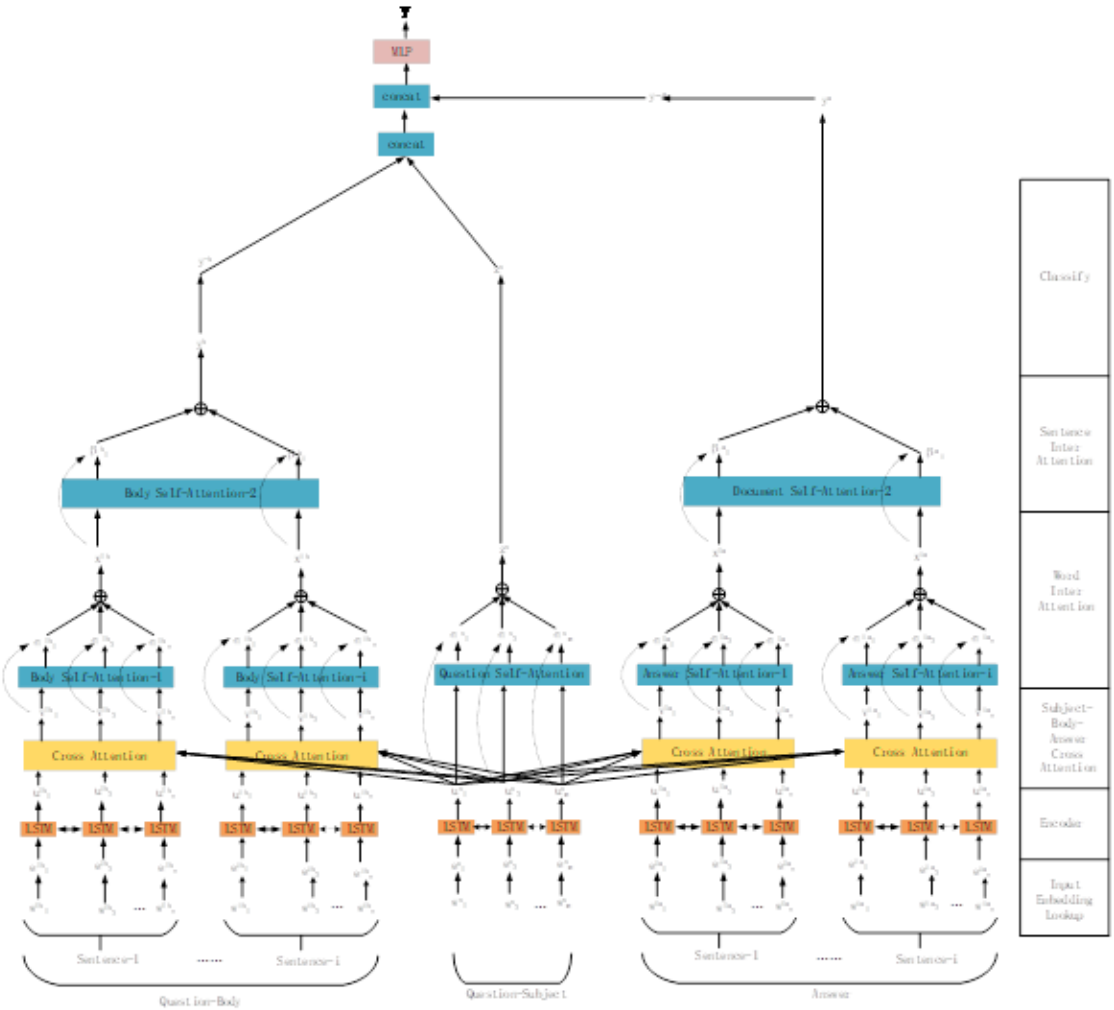

Figure 4.1: Overview of our proposed Himu-QAAN model.

objective, the deep intuition in this model is that the ratio of word-to-word co-occurrence probability has the potential to encode the meaning of some natural language form. During the past two years, some researchers have also proposed some more complicated word embedding models [60, 15].

In this work, various word-level embedding techniques are applied to form a special word-level embedding for answer selection task. Different from other corpora, such as news, governmental documents, reports, the corpus composed of materials from community usually has the following characteristics: the words are more colloquial, spelling errors may occur frequently, and emojis may appear.

To address these issues, first, GloVe is used to obtain the vector representation of most words. It is trained on the extracted Yahoo! Answers corpus But only using GloVe representation is not sufficient for comprehensive and accurate representations.

The reason is that, although the pre-trained GloVe "dictionary" is huge, it contains millions of words. However, there are sometimes a few words in the training set that are not in the GloVe vocabulary. Such words are called vocabulary (OOV) words. GloVe handles these OOV words by simply assigning them some random vector values. If left uncorrected, this random allocation will eventually confuse the obtained representation, resulting in reduced model performance. In order to solve the OOV problem, another embedding that can handle OOV words is used

in this research, which is character-level embedding [36] proposed by Kim et al. Character-level embedding can not only solve OOV problems, but also provide more benefits.

Another benefit is that it is very suitable for misspelled words, emojis and new words (for example, in 2018, the Oxford English Dictionary introduced more than 1,400 new words such as idiocracy, trapo, statocracy, bongga, boba tea, etc.) In addition, it also performs better than word2vec and GloVe when processing words that are not used frequently.

Therefore, in this work, three different modules of word embeddings are trained respectively, and then concatenated to form the final representation. Specifically, the word-level embedding comprises of the following three different modules: GloVe word representation, character embedding, and the syntactic features based on one-hot encoding (proposed by [18]).

The concatenation of the three embeddings provide various advantages. Firstly, the relationship of words is captured more accurately and precisely when the embedding is trained on the domain-specific corpus since the texts in CQA are mainly different in grammar and spelling from News and reports. Also, it has been proved that character embedding is effective for OOV (out-of-vocabulary), especially for CQA tasks. Furthermore, syntactic features based one-hot encoding provide more grammar information for better query representation.

For the task of answer selection, the question subject, the question subject, and the answer must be reasonably represented. First of all, the word set of all the candidates of question subjects is defined as $\{w_t^{subject}\}_{t=1}^{g}$, where g is the length of each question subject. Secondly, the word set of all the candidates of question bodies is defined as $\{w_t^{body}\}_{t=1}^{m}$, where m is the length of each question body. Let L be the number of sentences which are divided from the answers, and $\{w_t^{ianswer}\}_{t=1}^{n}$ be the words in sentence i in answer, where n represents the length of each sentence. Therefore, each candidate question-subject, question-body and answer can be represented as three vectors, $\{e_t^{subject}\}_{t=1}^{g}$, $\{e_t^{body}\}_{t=1}^{m}$ and $\{e_t^{ianswer}\}_{t=1}^{n}$, respectively.

4.4 Encoder

Long and short-term memory networks (commonly known as "LSTM") are a special type of RNN that can learn long-term dependencies, which were introduced by Hochreiter & Schmidhuber (1997). Based on this, many researchers have made considerable improvements, refinements and extensions in their subsequent work. Most of these works excel on a wide variety of issues and are now widely used. The main reason that researchers proposed this network is to avoid long-term dependence problem that RNN has. Compared with RNN, LSTM also has a chain structure, but the repeating module has a different structure. Moreover, LSTM is not only one neural network layer, but four nets interact in a very special way.

A typical LSTM network consists of different memory blocks called units. These units consist of the memory part of the LSTM unit and three "regulators" (often called gates). These three regulators are an input gate, an output gate, and a forget gate, which together form the internal information flow of the LSTM unit. LSTMs have some variant structures, that is, the unit does not have one or more of these gates, and there may be some other gates. These cells may have two states: cell states and hidden states. The biggest contribution of the cell state to the entire network is that it can carry relevant information during the entire sequence processing. Therefore, even information from earlier times can enter later time steps, reducing the impact of short-term memory.

Bidirectional LSTM is an extended variant of typical LSTM, which can enhance the performance of the model on sequence problems. Intuitively, a bidirectional LSTM is a double-layer structure formed by replicating a typical LSTM. When the network is running, the input sequence is provided as input to the first layer, the same as a typical LSTM, and an inverse copy of the input sequence is provided to the second layer.

In this work, for the encoder layer, bidirectional LSTM are used as encoders to convert the interaction among question-subject, question-body, and answer into coded form based on the temporal dependency, as shown in Fig.4.1. A H-dimensional contextual representation for per word is obtained by concatenating the output of

the two layers whose directions are different.

4.4.1 Question-Subject-Encoder

In most of the cases, question-subject consists of only one sentence, as simple as a title. This the why for question-subject, a simple bidirectional LSTM is used as the encoder. The input is embedding of question-subject, denoted as $\{e_t^{subject}\}_{t=1}^{g}$. The outputs are returns from the bidirectional LSTM, denoted as $U^{subject} = \{u_t^{subject}\}_{t=1}^{g} \in R^{g \times H}$.

The question-subject-encoder can be summarized as the following Equation (4.1):

$$u_t^{subject} = BiLSTM_{subject}(u_{t-1}^{subject}, e_t^{subject}) \tag{4.1}$$

4.4.2 Question-Body-Encoder

In Community Question Answering, question-body in most of the cases contain more than one sentences. Hence, first, to complete the encoding of question-body, firstly, each sentence should be encoded separately. Based on this, sentence-level bidirectional LSTM encoders are used to encode question-body with multiple sentences. The processing procedure is described as follows.

For each sentence i in question-body, the word embedding of sentence i is de-

noted as $\{e_t^{ibody}\}_{t=1}^{m}$. Then the encoding of sentence i is taken as the input and passed through the bidirectional LSTM. The output feedback by the network is the contextual word representation, denoted as $U^{ibody} = \{u_t^{ibody}\}_{t=1}^{m} \in R^{m \times H}$. Given an answer that contains P sentences, after encoding, the answer can be indicated as $\{U^{1body}, U^{2body}, \ldots, U^{Pbody}\} \in R^{P \times m \times H}$. Therefore, the mechanism of question-body-encoder can be demonstrated as Equation (4.2):

$$u_t^{ibody} = BiLSTM_{answer}(u_{t-1}^{ibody}, e_t^{ibody}) \tag{4.2}$$

4.4.3 Answer-Encoder

In Community Question Answering, answer is usually composed of multiple sentences, especially the best answer, which usually is even longer. Therefore, the same as question-body encoding, sentence-level LSTM is again used to encode answer. For each sentence i in answers, the word embedding of sentence i, denoted as $\{e_t^{ianswer}\}_{t=1}^{n}$, is taken as the input of the network. And then the encoding is passed through the bidirectional LSTM to obtain the output, which is the contextual word representation, denoted as $U^{ianswer} = \{u_t^{ianswer}\}_{t=1}^{n} \in R^{n \times H}$. Based on the above description, given an answer that contains L sentences, after encoding, the answer can be indicated as $\{U^{1answer}, U^{2answer}, \ldots, U^{Lanswer}\} \in R^{L \times n \times H}$.

The answer-encoder can be illustrated as Equation (4.3) :

$$u_t^{ianswer} = BiLSTM_{answer}(u_{t-1}^{ianswer}, e_t^{ianswer}) \tag{4.3}$$

4.5 Cross Attention

As its literal meaning shows, "attention" means focusing on something and getting more attention. Attention mechanism is one of the most influential ideas in the deep learning community in recent years. The attention mechanism in deep learning distributes attention to certain factors when processing data based on the concept of guided attention. Attention is an integral part of the network architecture and is responsible for managing and quantifying inter-dependencies. The operation performed by the attention component on each word in the output sentence is to map the key words and related words in the input sentence and assign higher weight to these words, thereby improving the accuracy of the output prediction. Attention was originally proposed to solve the main potential problems of the Seq2Seq model. The biggest disadvantage of the Seq2Seq method is that it needs to be able to compress all necessary information of the source sentence into a fixed-length vector. However, the process of compressing all the information of the input source sentence into a fixed-length and then obtaining it by the decoder has led to the performance degradation of Seq2Seq model when processing long sentences. Attention was originally designed in the context of neural machine

translation using the Seq2Seq model, which is an improvement on encoder-decoder-based neural machine translation systems in natural language processing. Later, it has become a successful solution for different tasks, such as image capture and speech recognition.

The attention mechanism assumes that the first word of the source sentence may be highly related to the first word of the target sentence. When attention predicts an output word, it uses only the most relevant information set in the input, not the entire sentence. In the attention mechanism, the encoder works the same as the encoder in Seq2Seq, but the hidden state of the decoder is calculated using the context vector, the previous output, and the previous hidden state. That is, the context vector is calculated as a weighted sum of the annotations generated by the encoder. Attention scores, the weights of hidden states when calculating context vectors, indicate the importance of a given annotation in determining the next state and generating output words.

Since the attention mechanism was proposed in 2017, many scientists have continued to improve on this basis. In this model, many improved attention networks are also used.

4.5.1 Cross Attention between Question-Body and Question-Subject

For the question selection task of this work, fusing question-subject with question-body are very important for exploring the interaction between question-subject and question-body.

The cross attention mechanism used in this research was proposed in [86, 99], which has been proven to be the best model for reading comprehension task, can achieve state-of-the-art performance. The research results [86] prove that in order for the model to achieve better results on the question and answer data set, it is necessary to ensure the robustness of distracting sentences, and only through learning heuristics of context and type matching.

This Cross Attention layer is used to fuse the words information of question-subject with words information of question-body, which is responsible for computing the relevance of each word in question-subject in regard to each word in question-body.

The attention layer uses the contextual embeddings generated by the encoders to compute the relevance between each pair of question-subject and question-body. Specifically, the bidirectional attention mechanism proposed in [66] is used to calculate the relevance of question-subject words with regard to question-body words, and vice-versa.

[66] obtains a question-aware contextual representation using Bi-Directional Attention Flow (BiDAF). Question-aware contextual representation is the interaction between a given paragraph and a question, which can be understood as embedding a question in a paragraph, which is to some extent equivalent to an encoding technique. [66] is an improvement over the existing attention mechanism.

The author of the paper summarizes the attention mechanism used in reading comprehension, which has three characteristics: (1) Attention weights usually summarize the context into a fixed-length vector, and then extract strong relevant information from the context to answer the question; (2) In the text domain, the attention weight is usually dynamic in time, where the attention weight at the current time step is a function of the previous time step participation vector; (3) it is usually a one-way attention from the problem to the text weight.

In this work, the bidirectional attention mechanism consists of two parts, *Subject2Body* and *Body2Subject*. *Subject2Body* is attention mechanism from question-subject to question-body, and *Body2Subject* is attention mechanism from question-body to question-subject. By computing the similarity between question-subject and question-body, a matrix is obtained, which can be denoted as $S_{body \sqcap subject} \in R^{g \times m}$. Then a softmax function is used for normalization over each row and column. Two similarity matrixes, $\bar{S}_{Subject2Body} \in R^{g \times m}$ and $\bar{S}_{Body2Subject} \in R^{g \times m}$, are generated after normalization, respectively. Finally, two attention matrices,

$A_{subject2Body} \in R^{g \times H}$ and $A_{Body2Subject} \in R^{g \times H}$, are obtained by following the below computation.

Let $s_{x,y} \in R$ be an element in similarity matrix $SBS \in R^{g \times m}$, where row represents question-subject, column represents question-body. Given inputs $U^{subject}$ and U^{body}, the final outputs are $V^{subject} = \{v_t^{subject}\}_{t=1}^{g}$.

The computation process can be described using the following Equations(4.4-4.9):

$$s_{x,y} = w_{subject}^T \cdot [u_x^{subject}; u_y^{body}; u_x^{subject} \odot u_y^{body}] \tag{4.4}$$

$$\bar{S}_{Subject2Body} = softmax_{row}(SBS) \tag{4.5}$$

$$\bar{S}_{Body2Subject} = softmax_{col}(SBS) \tag{4.6}$$

$$A_{Sbuject2Body} = \bar{S}_{Subject2Body} \cdot U^{Body} \tag{4.7}$$

$$A_{Body2Subject} = \bar{S}_{Subject2Body} \cdot \bar{S}_{Body2Subject}^T \cdot U^{subject} \tag{4.8}$$

$$V^{iSubject} = [U^{subject}; A_{Subject2Body}; U^{subject} \odot A_{Subject2Body}; U^{subject} \odot A_{Body2Subject}] \in R^{g \times 4H} \tag{4.9}$$

4.5.2 Cross Attention between Question-Body and Answer

This part focuses on fusing question-body with answers. Specifically, the cross attention mechanism is used to compute the relevance of each word in question-body regarding each word in answers. Also, the technique proposed in [86, 99] is applied again, because this is a model that achieves state-of-the-art performance on reading comprehension task, as mentioned before. The relevance of each question-body and answer pair is computed using the contextual embeddings generated by this technique. Moreover, the bidirectional attention mechanism proposed in [66] is applied again, to capture the relevance of question-body with regard to answer, and vice-versa. The two parts are denoted as *Answer2Body* and *Body2Answer*.

Furthermore, the similarity between question-body and answer is calculated to get the similarity matrix, denoted as $S_{body \sqcap answer} \in R^{n \times m}$. Then again, the softmax function is used for normalization over each row and column. Two similarity matrices are generated after normalization, they are $\bar{S}_{Answer2Body} \in R^{n \times m}$ and $\bar{S}_{Body2Answer} \in R^{n \times m}$. Finally, follow the below computation to get the two attention matrixes, $A_{Answer2Body} \in R^{n \times H}$ and $A_{Body2Answer} \in R^{n \times H}$.

Let $s_{x,y} \in R$ be an element in similarity matrix $SBA \in R^{n \times m}$, where row indicates answer, column indicates question-body.

Given inputs $U^{ianswer} \in \{U^{1answer}, \ldots, U^{Lanswer}\}$ and U^{body}, the final outputs are $V^{ianswer} = \{v_t^{ianswer}\}_{t=1}^{m} \in V^{1answer}, \ldots V^{Lanswer}$. The computation process can be denoted as the following Equations (4.10 - 4.16):

$$s_{x,y} = w_{answer}^T \cdot [u_x^{ianswer}; u_y^{body}; u_x^{ianswer} \odot u_y^{body}] \tag{4.10}$$

$$\bar{S}_{Answer2Body} = softmax_{row}(SBA) \tag{4.11}$$

$$\bar{S}_{Body2Answer} = softmax_{col}(SBA) \tag{4.12}$$

$$A_{Answer2Body} = \bar{S}_{Answer2Body} \cdot U^{body} \tag{4.13}$$

$$A_{Body2Answer} = \bar{S}_{Answer2Body} \cdot \bar{S}_{Body2Answer}^T \cdot U^{ianswer} \tag{4.14}$$

$$V^{ia} = [U^{ia}; A_{Ans2Body}; U^{ia} \odot A_{Ans2Body}; U^{ia} \odot A_{Body2Ans}] \in R^{n \times 4H} \tag{4.15}$$

$$V^{ib} = [U^{ibody}; A_{Body2Sub}; U^{ibody} \odot A_{Body2Sub}; U^{ibody} \odot A_{Sub2Body}] \in R^{m \times 4H} \tag{4.16}$$

4.6 Question-Body Inner Attention

In this layer, the self-attention mechanism proposed in [45] is used to fix question-subjects of different lengths to a uniform length. The self-attention mechanism allows extracting different convenient information of a sentence to form multiple vector representations. This is a new method for obtaining interpretable sentence embedding, which is realized by self-attention. Instead of vector 2D matrix is used for embedding.

Since the significance of a word in a document varies from document to document, and usually is determined by the context where the words occur. Hence, the self-attention technique assigns higher weights to the words that play more important role in questions, which ensures that the question representation is composed of features from more important words.

Let A be the dimension of question representation set. Given a specific feature of question-subject $U^{subject} = \{u_t^{subject}\}_{t=1}^{g}$ as input, a set of representation $z^s \in R^H$ is generated by this layer, the details are illustrated in the following Equations (4.17 - 4.19):

$$c_t^{subject} = w_{subject}^T(tanh(W_{subject}u_t^{subject}) \tag{4.17}$$

$$\alpha_t^{subject} = \frac{(exp(c_t^{subject}))}{\sum_{j=1}^{g} exp(c_j^{subject}))} \tag{4.18}$$

$$z^s = \sum_{t=1}^{g} \alpha_t^{subject} u_t^{subject} \tag{4.19}$$

4.7 Hierarchical Inner Attention

Answers often are longer in length than questions, hence, commonly only some parts of the whole answer are relevant to the question. In some cases, merely a few sentences can provide useful information. Even in each sentence, different words show various relevance to the question. Moreover, different answers often have different lengths, therefore, it is necessary to use a mechanism to obtain documents with fixed length. In this research, aim to fix document into unified length, a two-level inner attention mechanism proposed in [96, 110] is applied for document representation. This Inner Attention directly links the relationship between any two words in a sentence through a calculation step in the computation process, so the distance between long-distance dependent features is greatly shortened, which is conducive to effectively using these features. The first layer provides an effective way for lexical level representation in sentences, while the second layer is to obtain sentence-level representations.

Level-1 Attention: This layer applies attention on words in sentence. Each

sentence is encoded independently at word-level, so that a representation of each sentence with fixed dimension is obtained. This layer computes the importance of each word in the sentence, and then generates a collection of sentence representation based on the attention mechanism. Given a sentence i in answer A, this layer uses the $V^{ianswer} = \{v_t^{ianswer}\}_{t=1}^{n} \in R^{n \times 4H}$ as input, which is the output vector of cross attention layer. The output of Level-1 Attention is the representation of each sentence $x^{ianswer} \in R^{4H}$. The details is demonstrated as the followings (4.20 – 4.22):

$$c_t^{ianswer} = w_{a1}^T(tanh(W_{a1}v_t^{ianswer}) \tag{4.20}$$

$$\alpha_t^{ianswer} = \frac{exp(c_t^{ianswer})}{\sum_{j=1}^{n} exp(c_j^{ianswer})} \tag{4.21}$$

$$x^{ianswer} = \sum_{t=1}^{n} \alpha_t^{ianswer} u_t^{ianswer} \tag{4.22}$$

Level-2 Attention: This layer is for the answer representation on sentence-level. When computing the similarity, this layer allows the sentences that are more relevant and more informative to the question to gain higher attention. This layer takes the sentence representation $\{x_i^{ianswer}\}_{i=1}^{L}$ as input, and returns answer level vector $y^{answer} \in R^{4H}$, denoted as the following Equation (4.23 - 4.25):

$$b_i^{answer} = w_{a2}^T(tanh(W_{a2}z^{ianswer}) \tag{4.23}$$

$$\beta_i^{answer} = \frac{exp(b_i^{answer})}{\sum_{j=1}^{L} exp(b_j^{answer})} \tag{4.24}$$

$$y^{answer} = \sum_{j=1}^{L} \beta_j^{answer} z^{janswer} \tag{4.25}$$

Similarly, we can obtain Question-subject Hierarchical Inner Attention.

$$y^{body} = \sum_{j=1}^{P} \beta_j^{body} z^{jbody} \tag{4.26}$$

4.8 Prediction Layer

This layer is for the final prediction. Because the dimensions of y^{answer} and y^{body} are 4 times higher than the dimension of z^s, the answer representation y^{answer} is passed through a feedforward neural network for dimension reduction. After dimension reduction, $y^{(-answer)} \in R^H$ is obtained.

Similar, the question-body representation $y^{(-body)} \in R^H$ can be obtained. And then, $y^{(-answer)}$, $y^{(-body)}$ and z^s are concatenated to get the final representation p, which then is passed through a two-layer feedforward neural network. Finally, the probability distribution $Pr(y|Q, A)$ is computed in the last layer by a softmax

function. The softmax function is demonstrated as the Equation (27):

$$S_j = \frac{e^{\alpha_j}}{\sum_{k=1}^{T} e^{\alpha_k}} \tag{4.27}$$

As shown in (26), because e^x is always greater than 0, the numerator is always a positive number and the denominator is the sum of multiple positive numbers, so the denominator must also be a positive number. Therefore, S_j is positive, and the range of the value is (0,1).

When the model is used for testing instead of training, a sample passes through the softmax layer and outputs a $T*1$ vector. Then the index of the number with the largest value in this vector is taken as the prediction label for this sample. Softmax is usually used in the multi-classification process. Softmax maps the output of multiple neurons into the (0,1) interval, which can be interpreted as a probability to perform multi-classification. When the output node is finally selected, the node with the highest probability (that is, the value with the highest value) is selected as the prediction target.

The prediction process is demonstrated as the following Equations (28-31):

$$y^{-answer} = w_{d3}^{T} y^{answer} + b_{d3} \tag{4.28}$$

$$y^{-body} = w_{d4}^{T} z^{body} + b_{d4} \tag{4.29}$$

$$p = z^s concaty^{-body} concaty^{-answer} \tag{4.30}$$

$$Pr(y|Q, A) = MLP(p) \tag{4.31}$$

4.9 Dataset

Three datasets are used in this research, two of them are SemEval Dataset, the other is Yahoo! Answers dataset. In 1997, Senseval was established, and in 1998, 2001 and 2004, Senseval-1, 2, and 3 were successfully evaluated. The first three evaluations (Senseval-1 to Senseval-3) focused on word sense ambiguity elimination. Afterwards, due to the increasing tasks of semantic analysis in addition to word sense disambiguation in Senseval, the Senseval committee decided to change the evaluation name to an international semantic evaluation (SemEval), and organized SemEval2007 evaluation in 2007, its scale is unprecedented.

SemEval (Semantic Evaluation) is a series of international natural language processing (NLP) research workshops dedicated to promoting the latest developments in semantic analysis and helping to create high-quality annotated data sets to address a series of increasingly severe natural languages Semantic issues. Since then, the SemEval community has decided to hold an annual evaluation seminar together with the SEM conference, which showcases and compares the computa-

tional semantic analysis systems designed by different teams. But not every year's workshop share a set of tasks.

This article selects the datasets provided by SemEval 2015 and SemEval 2017 tasks. SemEval2015's task focuses on text similarity and question answering, time and space, emotion, word sense disambiguation and induction, and learning semantic relationships, so this data set is very suitable for this research on answer selection in community question and answer. SemEval 2017's task focuses on the semantic comparison of words and text, detecting emotions, humor and truth, and parsing the semantic structure. This data set is also very suitable for the research of this research.

The two corpus that used to train and evaluate our model are CQA datasets SemEval2015 and SemEval2017. SemEval2016 is not chosen in this experiment, because SemEval2017 is an updated version of SemEval2016, containing the same details that SemEval2016 has. There are two parts in the datasets, a list of questions and answers to the questions. Each question comprises a brief title and a more informative description. The statistics of the corpus is shown in Table 4.1 and Table 4.2.

The Yahoo! Answers dataset in this paper is extracted from the well-known question answering community Yahoo! Answers. Each category is sorted based on the number of questions. 60 categories are selected with the largest number

	Train	**Dev**	**Test**
Number of Questions	2376	266	300
Number of Answers	15013	1447	1793
Average Length of Subject	6.36	6.08	6.24
Average Length of Body	39.26	39.47	39.53
Average Length of Answer	35.82	33.90	37.37

Table 4.1: Statistical Information of SemEval2015 Corpus.

	Train	**Dev**	**Test**
Number of Questions	5124	327	293
Number of Answers	38638	3270	2930
Average Length of Subject	6.38	6.16	5.76
Average Length of Body	43.01	47.98	54.06
Average Length of Answer	37.67	37.30	39.50

Table 4.2: Statistical Information of SemEval2017 Corpus.

of questions, and the number of questions in these categories is more than 1000. All the samples are question-answer pairs to ensure that each question have a best answer. Also, for each question, our dataset provides several candidates that

Number of Questions	200,998
Number of Answers	1,848,441
Number of Best Answers	201,075
Number of classes	60
Number of Answers per Question	9.405

Table 4.3: Statistical Information of unpreprocessed Yahoo!Answers Corpus.

Number of Questions	60,000
Number of Answers	300,000
Number of classes	60

Table 4.4: Statistical Information of the original Yahoo!Answers Corpus.

might be relevant, or candidates that are clearly incorrect answers. Therefore, 1000 question-answer pairs are randomly selected from 60 categories are selected. According to the relevance of answers, each sentence contains 5 question-and-answer pairs, so that the final formation is represented as a set of triple elements (Question, Answer, Label). 70% of samples are randomly selected as the training set, 20% as the testing set, and 10% for validation. The statistics of the corpus is shown from Table 4.3 to Table 4.5.

	Training set	Testing set	Validation set
Number of questions	42,000	12,000	6,000
Average length of question	10.26	10.08	11.26
Average length of answer	42.38	41.26	40.68

Table 4.5: Statistical Information of the extracted Yahoo!Answers Corpus.

The distribution of percentage of answers with regard to the length of answers (number of words) in is demonstrated in Fig 2. It can be easily seen from Fig.6.1 that the proportion of answers with less than 50 words is very small, 5%. A vast amount of answers in the dataset contain 100-200 words, and a high proportion contains more than 200 words. Therefore, it is feasible to divide each answer into various fragments.

According to the types of questions in community Q & A, the questions can be divided as follows:

- Factoid questions: WH questions, such as when / who / where, etc .;
- Yes/No question: such as, Is Toronto the capital of Canada?
- Comparative question: Which city is larger, Toronto or Ottawa?
- Opinion question: What is your opinion about Donald Trump?

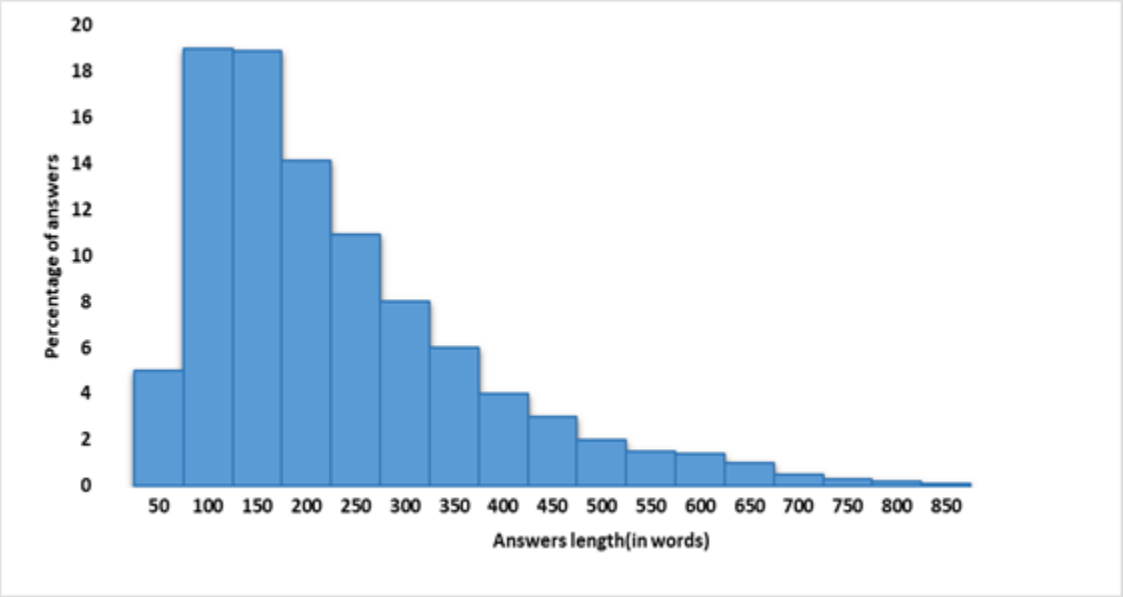

Figure 4.2: Percentage of Answers vs Number of Words.

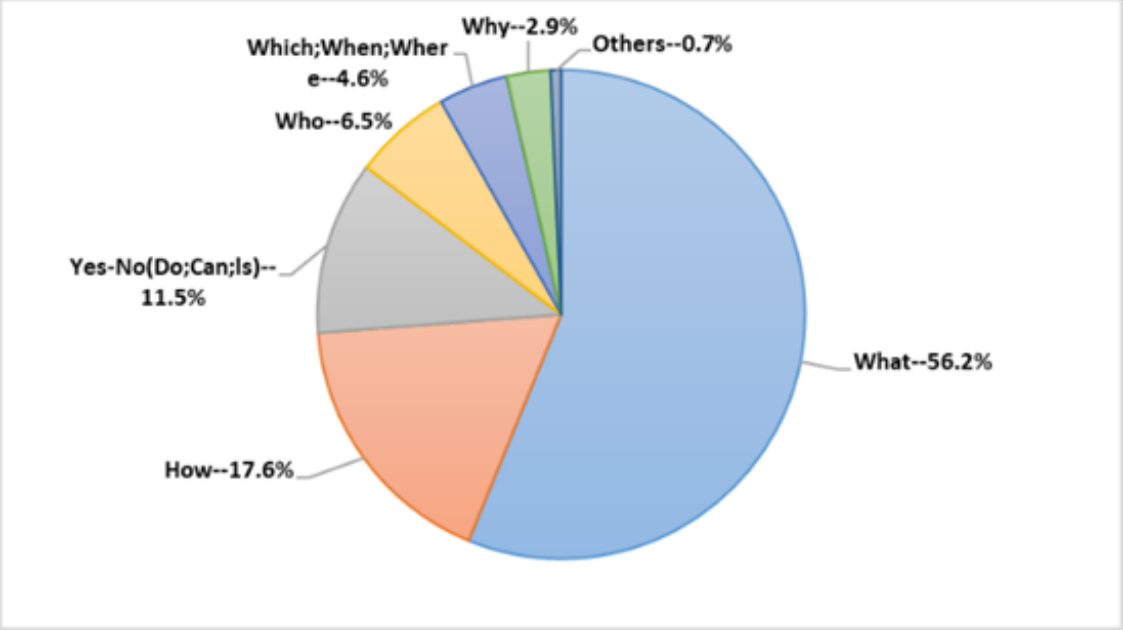

Figure 4.3: Percentage of Questions by Type.

- Cause/result questions: how / why / what ... for, etc.

The objective questions such as Yes/No type of question and Comparative question are based on factual questions (for example, the factoid question "What is the capital of Canada?" is the basis of answer to Yes/No question "Is Toronto the capital of Canada?". For the comparative question "Which city is larger, Ottawa or Toronto?", the answers of factoid questions "How large is Ottawa? and How large is Toronto?" are the basis for answering the comparative one.). Therefore, question types can be simply divided two, factoid and non-factoid.

Fig 6.2 shows the percentage of different types of questions in our dataset. In essence, the questions of type "How" or "Why" are mostly non-factoid. These questions are open-ended, usually require detailed answers with comprehensive descriptions or reasonable arguments. Factoid questions usually have standard answers, while non-factoid questions usually do not have standard answers and can be discussed from different perspectives. Therefore, the answers to factoid questions are generally longer than non-factual questions.

As shown in Fig.6.2, the question denoted as type "What" are generally considered to be factoid which account for a large proportion in this dataset. But after analysis, the results illustrate that a large part of such questions is also non-factoid. Examples of such questions include "What is the outlook of natural language processing?". It can be inferred that the expected answers to this question must be

descriptive answers of considerable lengths. Based on the above analysis, it can be deduced that most of the question and answer pairs in the Yahoo! Answers dataset are non-factoid.

4.10 Training and Hyperparameters

In this section, the details of pre-processing are described. Also, the details of optimizer and other hyperparameters are listed.

4.10.1 Pre-processing

Pre-processing is a key step in natural language processing tasks, and it is no exception to the task of answer selection for CQA. In this section, the methods used to clean up the text data set are introduced, thereby eliminating implicit noise and allowing informative featurization. Most text and document data sets contain many unnecessary words, such as stop words, misspellings, slang, etc.

Whether it is statistical and probabilistic learning algorithms, or deep learning algorithms, noise and unnecessary functions may adversely affect model performance. For text pre-processing procedure, the NLTK toolkit is exerted over each question and paired answers. The techniques and methods are used for pre-processing include tokenization (Tokenization is a pre-processing method that can decompose a text stream into words, phrases, symbols, or other meaningful ele-

ments called tokens)., capitalization conversion(usually convert each letter to lower case), stemming(consolidating different forms of the same word, such as the singular and plural of nouns, the past and -ing forms of verbs, etc.), removal of stop words such as “a”, “about”, “above”, “across”, “after”,“afterwards”, “again”,. . ., converting slang and abbreviation into formal language, noise removal such as punctuation and special characters, spelling correction, et al.

4.10.2 Word Embedding

After preprocessing, the datasets were trained with Glove to obtain 300-dimensioanl initialized word vectors. The representations for out-of-vocabulary words are set to zero.

4.10.3 Optimizer

In the field of deep learning, the choice of optimization algorithm is the top priority of a model. Even when the data set and model architecture are completely the same, the use of different optimizer is likely to result in completely different training effects.

- **Gradient descent**

  Gradient descent is one of the most widely used optimization algorithms in neural networks. In order to make up for the shortcomings of Vanilla gradient

descent, researchers have invented a series of variant optimizers including the most famous stochastic gradient descent (SGD).

Gradient descent means that, given the model parameters $\theta \in R^d$ and objective function $J(\theta)$, the algorithm is to minimize $J(\theta)$ by updating θ in the opposite direction of the gradient $\nabla_\theta J(\theta)$. The learning rate η determines the update step size at each moment. For each moment t, the following steps can be used to describe the gradient descent process:

1. Calculate the gradient of the objective function with respect to the parameters.

$$g_t = \nabla_\theta J(\theta) \tag{4.32}$$

2. Calculate the first and second order momentum based on the historical gradient.

$$m_t = \phi(g_1, g_2, \ldots, g_t) \tag{4.33}$$

$$v_t = \psi(g_1, g_2, \ldots, g_t) \tag{4.34}$$

3. Update the model parameters.

$$\theta_{t+1} = \theta_t - \frac{1}{\sqrt{v_t + \epsilon}} \tag{4.35}$$

Where, ϵ is a smooth term, to prevent the denominator from zero, usually can be defined as 1e-8.

- **Vanilla SGD**

  With the continuous development of deep learning, gradient descent has also produced different variant algorithms. Vanilla SGD is the simplest, without the concept of momentum. Also, the update step is the simplest, update the parameter (weight) in the direction of the gradient, that is:

$$W \leftarrow W - \eta \frac{\partial L}{\partial W} \tag{4.36}$$

  W is the weight parameter, L is the loss function, η is the learning rate, $\frac{\partial L}{\partial W}$ is the gradient (differential) of the loss function to the parameter.

  The main drawback of SGD is that the convergence speed is slow and may oscillate at the saddle point. Moreover, how to choose a reasonable learning rate is a major difficulty for SGD.

- **Momentum**

  SGD can easily fall into shock when it encounters a gully. Therefore, Momentum is introduced to accelerate the decline of SGD in the correct direction and suppress oscillation.

$$V_t \leftarrow \beta V_{t-1} - \eta \frac{\partial L}{\partial W} \tag{4.37}$$

$$W \leftarrow W + V_t \tag{4.38}$$

There is an additional V_t parameter here, which can be imagined as "direction speed", which will be related to the last update. If the last gradient is in the same direction as this time, $\mid V_t \mid$ (speed) will become larger and larger (representing Gradient enhancement), the update gradient of W parameter will become faster and faster. If the direction is different, $\mid V_t \mid$ will be smaller than the last time (gradient weakening), the update gradient of W parameter will become smaller, β can be imagined as air resistance or ground friction, usually set to 0.9.

- **AdaGrad**

  For the Optimizer, the learning rate (learning rate) η is very important. Too small will take too much time to learn. If it is too large, it may cause over-fitting and cannot be learned correctly. The learning rate of the previous Optimizer is a fixed value. , And AdaGrad is an optimizer that adjusts the learning rate according to the gradient.

$$W \leftarrow W - \eta \frac{1}{\sqrt{n+\epsilon}} \frac{\partial L}{\partial W} \tag{4.39}$$

$$n = \sum_{r=1}^{t} (\frac{\partial L_r}{\partial W_r})^2 \tag{4.40}$$

$$W \leftarrow W - \eta \frac{1}{\sqrt{\sum_{r=1}^{t} (\frac{\partial L_r}{\partial W_r})^2 + \epsilon}} \frac{\partial L}{\partial W} \tag{4.41}$$

In AdaGrad Optimizer, η is multiplied by $\frac{1}{\sqrt{n+\epsilon}}$ and then the parameter is updated, and a parameter of n appears, where n is the sum of the squares of all the previous gradient values, and the learning rate is adjusted using the squared sum of the gradient values learned earlier , ϵ is a smooth value, the reason for adding is to prevent the denominator from being 0, and the general value of ϵ is 1e-8.

When the early gradient is small, n is small, which can amplify the learning rate. When the gradient is larger in the later period, n is larger, which can constrain the learning rate, but the accumulation of the gradient square on the denominator will become larger and larger, making the gradient closer to 0, and the training will end. In order to prevent this, there will be development later Out of RMSprop Optimizer, the main is to change n into RMS (root mean square).

- **Adam**

  Adam is a first-order optimization algorithm that can replace the conventional stochastic gradient descent process. Adam can iteratively update neural network weights based on training data. Adam was originally proposed by Diederik Kingma of OpenAI and Jimmy Ba of the University of Toronto in the 2015 ICLR paper (Adam: Method for Stochastic Optimization). The algorithm is called "Adam", which is not an acronym, nor is it a person's name. Its name comes from adaptive moment estimation. Adam Optimizer can actually be seen as a combination of Momentum and AdaGrad, which were introduced earlier.

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1)\frac{\partial L_t}{\partial W_t} \tag{4.42}$$

$$v_t = \beta_1 v_{t-1} + (1 - \beta_2)(\frac{\partial L_t}{\partial W_t})^2 \tag{4.43}$$

  Like Momentum, the average value of the exponential decay of the past gradient is maintained, and like AdaGrad, the average value of the squared decay of the past gradient is stored.

  Then do deviation correction for m_t and v_t:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t} \tag{4.44}$$

$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{4.45}$$

So the update of the parameters can be expressed as

$$W \leftarrow W - \eta \frac{\hat{m}_t}{\sqrt{\hat{v}_t} + \epsilon} \tag{4.46}$$

In this work, when conducting experiments, Adam Optimizer is chosen as the optimization approach, with momentum coefficient 0.9, the second momentum coefficient 0.999.

4.10.4 Hyperparameters

The model is learned with initial learning rate $[10 \times 10^{-9}, 4 \times 10^{-5}, 1 \times 10^{-7}]$, L2 regularization parameters $[10 \times 10^{-6}, 4 \times 10^{-7}, 1 \times 10^{-7}]$, and batch size [64, 128, 256]. The parameters are selected with the best performance on validation set, and then performance of the model is evaluated on test sets.

4.11 Evaluation Metrics

In this research, three evaluation metrics, F1, accuracy (Acc), and MAP (Mean Average of Precision) are adopted to compare the performance of HASANAN and the baseline models.

4.12 Baselines

Seven answer selection models are used as comparison models for SemEval 2015 dataset. As shown in Table 4.6, (1), (2), (3), (4), (5), 6), (7) are the Baseline of Himu-QAAN. (1) and (2) are top systems from SemEval 2015. Compared to the model in this research which learns various important features automatically, Baseline (1), (2), (3), (4) highly reply on feature engineering. (3) uses thread level information for global inference, (6) is the top technique for Q & A task from SemEval 2017, and (5), (7), (8) are neural network based deep learning models.

To be specific, baseline (1) introduced their work of SemEval-2015 Task 3: Answer Selection in Community Question Answering. The researchers propose a model that evaluates the quality of answers by fusing multiple features, including word matching features (including cosine similarity, dependency similarity, word alignment and noun matching), special component features (including special symbols and specific Vocabulary), topic-based features (that is, the topic matching degree

of the question and answer), translation-based features (that is, each word in the question will be aligned with the word in the answer, and get the highest translation score, the feature value is translation Sum of scores) and non-text features (including the author of the question, the number of questions posted by the author, and the directory where the question is located).

Baseline (2) is an outstanding model in the 2015 SemEval answer selection challenge. The model participated in the 2015 SemEval English subtask A, English subtask B and Arabic task. Researchers have proposed two methods of integrated learning and hierarchical classification for the choice of answers for each task. In this study, bag-of-word features, lexical features, and non-text features are used. For Arabic tasks, features are extracted from Arabic data and English data translated from Arabic data.

Baseline (3) relys on thread level information to make global inference. It was evaluated on the benchmark data set of SemEval-2015 task 3. Baseline (4) consists of two novel joint learning models, which are online and integrate inference in learning.

Baseline (5) introduces the Bi-directional Gated Memory Network (BGMN) to model the interactions between question and answer. (5) was evaluated on SemEval-2015 Task 3 dataset.

Baseline (6) is for task 3 (community question and answer) of SemEval 2017, this

task contains three subtasks about the English corpus, namely subtask A: question-note similarity, subtask B: question-problem similarity and subtask C: Question-External comments are similar. For subtask A, (6) combines two different methods to express problem-comment pairs, that is, a supervised model using traditional features and convolutional neural networks. For subtask B, the excerpt information returned from the search engine is used to query the subject of the question. For subtask C, the comments are ranked by multiplying the probability of commenting on the pair of related questions by the reciprocal of the related question.

Baseline (7) proposed two models based on neural networks, which have different combinations of Convolutional Neural Networks, Long Short Term Memory and Conditional Random Fields. Extensive experiments were conducted on the data set released by the SemEval-2015 CQA shared task.

Baseline (8) is a deep learning based model, named Question Condensing Networks (QCN), which make use of the subject-body relationship of community questions.

Also, seven answer selection models are used as comparison models for Yahoo! Answer dataset. As shown in Tab7, from model (1) to model (7). The baseline models for SemEval 2017 are listed in Tab 8. (1), (2), and (3) are top systems from the SemEval 2017.

More detailed, baseline (1) is the KeLP system participating in the SemEval-

2017 Community Question Answering (CQA) task. The system is an improvement of the kernel-based sentence modeling proposed in the previous year's challenge. It is named KeLP because it is implemented in a kernel-based learning platform called KeLP. Baseline (1) ranks first in subtask A of SemEval task 3 and third in subtasks B and C, is the only system that appears in the top three of all English subtasks.

Baseline model (2) is proposed for SemEval-2017 task 3. Its essence is a ranking system, which can capture the semantic relationship between text pairs without word overlap. System (2) ranks second in subtask A and fifth in subtask B. Baseline (3) is the same with the SemEval 2015 baseline (6).

(4) to (6) are deep learning baselines, and (6) is the same with the SemEval 2015 baseline (8). (4) is the classic LSTM network. (5) is the LSTM based network applying on question subject and question body respectively. The final representations of questions are the concatenation of each part.

The baseline models for SemEval 2017 are listed in Tab 4.7. (1), (2), and (3) are top systems from the SemEval 2017.

More detailed, baseline (1) is the KeLP system participating in the SemEval-2017 Community Question Answering (cQA) task. The system is an improvement of the kernel-based sentence modeling proposed in the previous year's challenge. It is named KeLP because it is implemented in a kernel-based learning platform

called KeLP. Baseline (1) ranks first in subtask A of SemEval task 3 and third in subtasks B and C, is the only system that appears in the top three of all English subtasks.

Baseline model (2) is proposed for SemEval-2017 task 3. Its essence is a ranking system, which can capture the semantic relationship between text pairs without word overlap. System (2) ranks second in subtask A and fifth in subtask B. Baseline (3) is the same with the SemEval 2015 baseline (6).

(4) to (6) are deep learning baselines, and (6) is the same with the SemEval 2015 baseline (8). (4) is the classic LSTM network. (5) is the LSTM based network applying on question subject and question body respectively. The final representations of questions are the concatenation of each part.

4.13 Experiment on Datasets

To analyze the effectiveness of the proposed model, some traditional and state-of-the-art methods mentioned before are selected as baseline models. Also, two standard datasets above are chosen as evaluation corpus. Firstly, the following two evaluation metrics, F1, accuracy (Acc) are adopted to compare the performance of Himu-QAAN and the baseline models. Seven answer selection models are used as comparison models for SemEval 2015 dataset.

The results shown in Table 4.8 demonstrated that:

- Not all deep learning models can outperform conventional machine learning models. SVM classifier JAIST achieves better performance than deep learning model BGMN, 0.0173 in terms of F1, and 0.007 in terms of Acc. (Row 1 VS Row 5)

- Graph-cut and FCCRF outperform JAIST, and HITSZ-ICRC on these two evaluation metrics. Specifically, Graph-cut outperforms JAIST 0.0159 in terms of F1, and 0.0700 in terms of Acc. Graph-cut outperforms HITSZ-ICRC 0.0403 in terms of F1, and 0.0369 in terms of Acc. FCCRF shows better performances than JAIST, 0.0254 in terms of F1, 0.014 in terms of Acc. FCCRF shows better performances than HITSZ-ICRC, 0.0498 in terms of F1, 0.0439 in terms of Acc. These results prove that if used properly, heavy feature engineering models can achieve better performance than conventional machine learning models in answer selection task. (Row 3, Row 4 VS Row 1, Row 2)

- Furthermore, Graph-cut and FCCRF outperform BGMN on both F1 and Acc evaluation metrics. More detailed, Graph-cut outperforms BGMN 0.0332 in terms of F1 and 0.014 in terms of Acc. FCCRF outperforms BGMN 0.0427 in terms of F1 and 0.021 in terms of Acc. These results prove that if used properly, heavy feature engineering models can achieve better performance

than deep learning model in answer selection task. (Row 3, Row 4 VS Row 5)

- The experiment results show that the proposed model Himu-QAAN achieves the best performance on this dataset, outperforming the best baseline model (7) QCN by 0.71% in terms of F1, 1.41% in terms of Acc, respectively. (Row 8 VS Row 7) The self-attention mechanism helps Himu-QAAN to find the words in answer that are most relevant to the question. By using cross attention between question subject and answer, question body and answer, Himu-QAAN can successfully learn these crucial interaction features, extract the similarity between question and answer, also capture the semantics of the answer sentences.

The next step, the following three evaluation metrics, F1, accuracy (Acc) are adopted to compare the performance of HASAN and the baseline models on SemEval 2017 dataset. Seven answer selection models are used as comparison baseline models for SemEval 2017 dataset.

The results in Table 4.9 demonstrate that:

- Overall, the proposed model HASAN outperform all the baseline models. (Row 7 VS Row 1, Row 2, Row 3, Row 4, Row 5, Row 6)

  Compared to the best baseline (6) QCN, HASAN shows 1.11% better in terms

of MAP, 1.04% better in terms of F1, and 1.07% better in terms of Acc. (Row 7 VS Row 6)

- Also, the similar as the experiment results of SemEval 2015 dataset, it can be observed that deep learning models do not always achieve unsurpassable results. Baseline (1) KeLP, (2) Beihang-MSRA outperform two deep learning models (4) LSTM and (5) LSTM-subject-body in terms of MAP.

  Specifically, KeLP outperforms baseline (4) by 0.0211 in terms of MAP. (Row 1 VS Row 4)

  KeLP ourperforms baseline (5) by 0.0132 in terms of MAP. (Row 1 VS Row 5)

  Beihang-MSRA shows better performance than model (4) LSTM in terms of MAP by 0.0192. (Row 2 VS Row 4)

  Beihang-MSRA outperforms baseline (5) LSTM-subject-body by 0.0113 in terms of MAP. (Row 2 VS Row 5)

- Compared baseline (4) with baseline (5), the model treat question subject and question body separately outperform the model taking the subject and question as an entirety, i.e., model (5) LSTM-subject-body outperforms model (4) LSTM by 0.0079 in terms of MAP, 0.0009 in terms of F1, and 0.0159 in terms of Acc. (Row 4 VS Row 5)

This comparison proves that reasonably treat question body and question subject can enhance the performance for answer selection task.

- Moreover, compare baseline (5) with baseline (6), model (6) QCN outperforms model (5) by 0.0140 in terms of MAP, 0.0361 in terms of F1, and 0.0343 in terms of Acc. (Row 6 VS Row 4)

  This comparison illustrates that the mechanical connection of question-subject and question-body introduces too much noise information and redundancies, resulting in the performance degradation.

- The above results demonstrate that the question-object information could be an essential knowledge source for answer selection, and the proposed model makes effective use of this information.

The experiment results in Tab 4.10 demonstrate that:

- Baseline model (1) JAIST and baseline model (2) HITTZ-ICRC are two models with good performance on the dataset SemEval2015. The experiment results show that JAIST outperforms baseline (5) BGMN by 0.0196 in terms of MAP, 0.0211 in terms of F1, and 0.0231 in terms of Acc. In general, JAIST shows better performance than BGMN on three evaluation metrics, which confirms that not all deep learning models can outperform conven-

tional machine learning models. Conventional machine learning model can show significant performances in answer selection task. (Row 1 VS Row 5)

- The baseline model (6) ECUN combining convolutional neural network with supervised learning outperforms baseline (5) BGMN and baseline (2) HITZZ-ICRC on all three evaluation metrics.

  Specifically, ECUN outperforms baseline (5) BGMN by 0.0079 in terms of MAP, 0.0113 in terms of F1, and 0.0115 in terms of Acc. (Row 6 VS Row 5)

  ECUN outperforms baseline (2) HITSZ-ICRC by 0.0184 in terms of MAP, 0.0185 in terms of F1, and 0.0157 in terms of Acc. (Row 6 VS Row 2)

  The results demonstrate that the combination of conventional machine learning model and deep learning model can achieve better performance in answer selection task than models that only apply machine learning techniques or deep learning techniques.

- The baseline (3) Graph-cut and (4) FCCRF outperforms baseline (6) ECUN, (5) BGMN, (2) HITZZ-ICRC, and (1) JAIST on all three evaluation metrics.

  Specifically, Graph-cut outperforms baseline (1) JAIST by 0.0010 in terms of MAP, 0.0023 in terms of F1, and 0.0078 in terms of Acc. (Row 3 VS Row 1)

  Graph-cut outperforms baseline (2) HITZZ-ICRC by 0.0311 in terms of MAP, 0.0306 in terms of F1, and 0.0351 in terms of Acc. (Row 3 VS Row 2)

Also, FCCRF shows better performance than baseline (1) JAIST by 0.0141 in terms of MAP, 0.0196 in terms of F1, and 0.0228 in terms of Acc. (Row 4 VS Row 1)

FCCRF shows better performance than baseline (2) HITZZ-ICRC by 0.0442 in terms of MAP, 0.0306 in terms of F1, and 0.0351 in terms of Acc. (Row 4 VS Row 2)

These comparisons prove that heavy feature engineering models, if used properly can achieve better performance than conventional machine learning models in answer selection task.

- The baseline (7) CNN-LSTM-CRF outperforms baseline (3) Graph-cut, baseline (4) FCCRF, baseline (6) ECUN, (5) BGMN, (4) FCCRF, (2) HITSZ-ICRC, and (1) JAIST on three different evaluation metrics, MAP, F1, and Acc, respectively. (Row 7 VS Row 6, Row 5, Row 4, Row 3, Row 2, Row 1)

  The results demonstrate that a CNN, LSTM, CRF layer mixed model can achieve better performance on answer selection task than conventional machine learning models, neural network models, heavy featuring engineering models, and model that combines machine learning techniques and deep learning techniques, at least in the case of this research.

- Himu-QAAN outperforms other seven models in terms of three different eval-

uation metrics. (Row8 VS Row 7, Row 6, Row 5, Row 4, Row 3, Row 2, Row 1)

For instance, Himu-QAAN outperforms the most strongest baseline (7) CNN-LSTM-CRF by 0.0442 in terms of MAP, 0.0259 in terms of F1, and 0.0376 in terms of Acc. (Row 8 VS Row 7)

The results prove that paired answer can provide important information for answer selection, which makes the proposed model Himu-QAAN more powerful than machine learning models, heavy feature engineering models, models that combine various techniques, and model with different deep neural layers.

Moreover, by using cross attention between question and answer, Himu-QAAN can successfully learn these crucial interaction features and extract the similarity between question and answer, also capture the semantics of the answer sentences.

Furthermore, the self-attention mechanism helps the model to find the words in answer that are most relevant to the question.

These are the reasons that the proposed model achieves the best performance compared to the seven baselines.

4.14 Ablation study on SemEval 2017 dataset

In order to fully verify the validity of the proposed model, in addition to the comparison with the baseline models, ablation studies are implemented in this section.

First, six extra baselines on SemEval 2017 are implemented on the SemEval 2017 datase. The detailed experimental methods and description for comparisons as listed as followings.

(1) w/o task-specific word embeddings

word embeddings are initialized with the 300-dimensional GloVe word vectors trained on Wikipedia 2014 and Gigaword 5.

(2) w/o character embeddings

word level embeddings are only composed of 600-dimensional GloVe word vectors trained on the domain-specific unannotated corpus.

(3) without cross-attention

To evaluate the effect of using cross-attention mechanism on model performance, we evaluate the performance of a variant of Himu-QAAN that does not use cross-attention between question and answer words. This model uses an inner attention over question words, and a hierarchical inner attention over answer words and sentences.

(4) Cross Attention between subject-body concatenation and answer

Question-subject and question-body is concatenated, and then the entirety is used as input for cross attention.

(5) Only Cross Attention between subject and answer

Only takes question-subject as input, and then uses question-subject and answer for cross attention.

(6) Only Cross Attention between body and answer

Only takes question-body as input, and then uses question-body and answer for cross attention.

After comparing SemEval 2017 dataset with SemEval 2015, SemEval 2017 is chosen for two reasons. First, it is larger than SemEval 2015, which means technically SemEval 2017 is more likely to strengthen the training of data-driving model. Secondly, it is deduced that the types of questions and the paired answers of SemEval 2017 dataset might demonstrate the improvement process of the proposed model more precisely. The results of the ablation study on SemEval 2017 dataset are listed in Table 4.12.

The results demonstrate that using task-specific embeddings and character embeddings both contribute to model performance. This is because CQA text is non-standard. The text of CQA usually contain a large number of oral descriptions

and informal expressions. Task-specific embeddings are obtained by training specifically on the CQA corpus. Therefore, semantic features and syntactic features can be captured more accurately. Based on this, task-specific embeddings exhibit the most outstanding performance.

To be specific, the comparison between model (1) and (2) proves that there are quantities of informal language usage, such as abbreviations, typos, emoticons, and grammatical mistakes. Using task-specific embeddings and character embeddings can help to attenuate the OOV problem.

From the comparison between (4) and (5), it is illustrated that cross attention between subject-body concatenation and answer can achieve better performance than only using cross attention between subject and answer. The reason is that the information from question-subject is limited, cannot provide the deep semantic representation of question as good as subject-body concatenation.

Moreover, the comparison between (4) and (6) demonstrates that only using cross attention between question-body and answer can achieve better performance than using subject-body concatenation. This is most likely caused by unnecessary noises of subject-body concatenation. These noises result in bad interference to the accuracy of the model, so that the performance of model (4) is not as good as model (6).

Furthermore, as shown in Table 13, model (6) which applies cross attention be-

tween question-body and answer outperforms model (5) that applies cross attention between question-subject and answer. These results prove that question-body is more informatic than question-subject. Hence, by absorbing the useful semantic information provided by question-body, the performance of model is boosted.

Model (3) is the only model without using cross-attention mechanism, which shows the worst performance among these 7 models. These results again demonstrate that cross attention can capture the relationship between question and answer, and help assign different attention to different feature words. As a consequence, models with cross-attention can easily outperform model (3).

The proposed model Himu-QAAN studies the relationship of question-body, question-subject, and answer, fully uses word level attention and cross-attention to enhance the performance of answer selection. HASAN achieves the best performance among all the seven models.

4.15 Ablation study on Yahoo! Answer dataset

In order to fully verify the validity of the proposed model, furthermore, in addition to the comparison with the baseline models, six extra baselines on Yahoo! Answers dataset are implemented, as listed in Tab 4.13. The results are listed in Table 4.14.

From the results in Table 7.6, it can be demonstrated that using task-specific embeddings and character embeddings both contribute to more precise semantic

meaning extraction. This is because the expressions are informal, and sometimes non-standard.

The comparison between model (1) and (2) proves that there are quantities of informal language usage, such as abbreviations, typos, emoticons, and grammatical mistakes. Also, using task-specific embeddings and character embeddings can help to attenuate the OOV problem.

Model (3) is the only model without using cross-attention mechanism, which shows the worst performance among these 6 models. This is because cross-attention can capture the relationship between question and answer, and assign different attention to different feature words. So that models with cross-attention can easily outperform model (3).

The comparison between (4) and (5) demonstrates that model using word inter attention outperform model using sentence inter attention. The reason is that word inter attention can capture the semantic information between words more efficiently, which also proves that attention between words is more powerful than sentence-level attention for answer selection tasks.

The Himu-QAAN model proposed in this study studies the relationship between questions and answers, making full use of word, sentence, and document-level hierarchical attention mechanisms to improve the performance of answer selection.

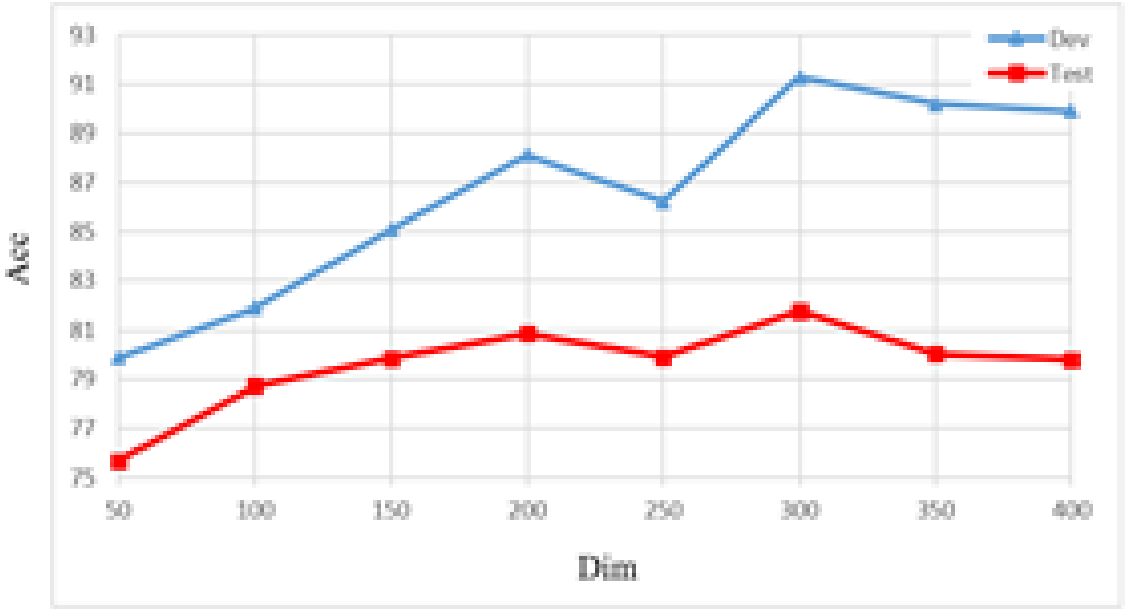

Figure 4.4: Results of the proposed model influenced by different hidden state dimensions of LSTMs.

4.16 Parameter Sensitivity

In this section, the impact of LSTM's hidden state dimension and answer sentence length on SemEval 2017 are evaluated, respectively.

The hidden state dimensions of LSTMs may impact the performance of models. Hence, it is necessary to investigate the impact. Fig 2 shows the model's achieved results for different dimensions. As shown in the figure, when the hidden state size is less than 300, the performance of model Himu-QAAN is increasing along with it. This trend indicates that a large hidden state size could enhance the performance

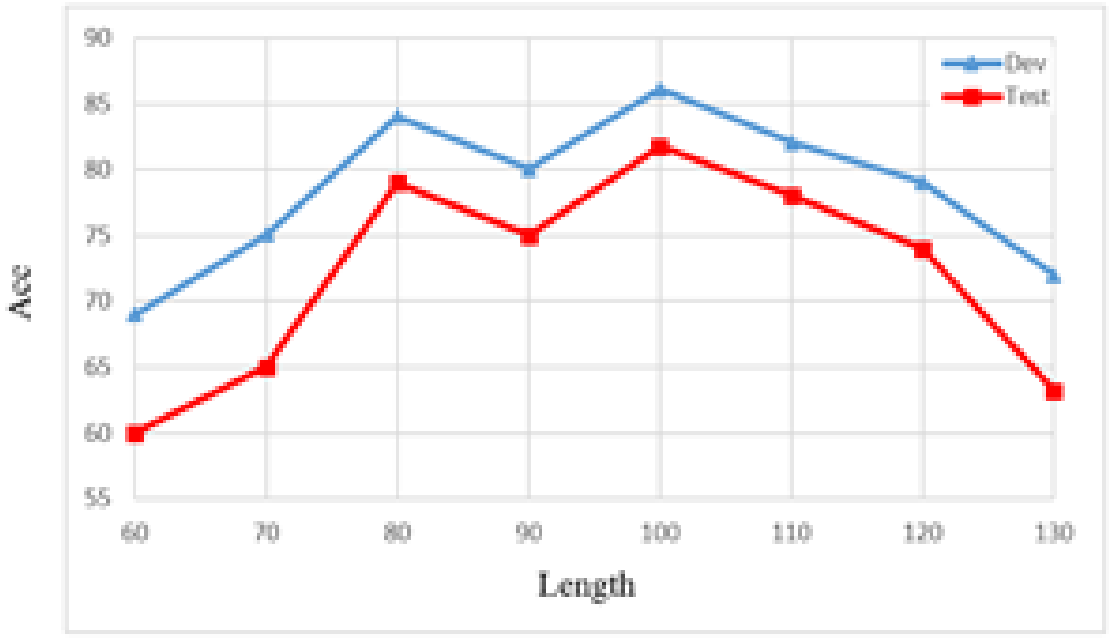

Figure 4.5: Results with answers that are truncated at different lengths.

of model Himu-QAAN. When the dimension reaches 400, however, the performance drops on both the dev and test sets. This may be due to a requirement of more data for fitting such a large number of parameters. In this research, the best result is acquired when the hidden state dimensions of the LSTMs are set to 300.

Further, the performance of model Himu-QAAN with answers that are truncated at different lengths is compared. As illustrated in Figure 4, model Himu-QAAN achieves the best performance at the truncated length of answers as 100. As mentioned before, the answer information for this task is usually a mixture of valuable information and other redundant information. Therefore, a shorter truncation may cause the useful information to be lost, while a longer truncation may

introduce more redundant information to aggravate the noise problem.

(1) JAIST	See Table 5.4.
(2) HITSZ-ICRC	See Table 5.4.
(3) Graph-cut	Proposed by [33]. It modeled the relationship between pairs of answers at any distance in the same question thread, based on the idea that similar answers should have similar labels.
(4) FCCRF	Proposed by [34]. It used locally learned classifiers to predict the label for each individual node, and applied fully connected CRF to make global inference.
(5) BGMN	See Table 5.4.
(6) ECUN	See Table 5.4.
(7) CNN-LSTM-CRF	Proposed by [92]. The question and its answers are linearly connected in a sequence and encoded by CNN. An attention-based LSTM with a CRF layer is then applied on the encoded sequence.
(8) QCN	Proposed by [89]. It is a Question Condensing Networks that use the similarity and disparity between question-subject and question-body for answer selection.
(9) Himu-QAAN (Our model)	

Table 4.6: Baseline models of the SemEval2015 dataset.

(1) KeLP	Proposed by [20]. It used syntactic tree kernels with relational links between question and answer, and standard text similarity measures linearly combined with the tree kernel.
(2) Beihang-MSRA	Proposed by [18]. It used gradient boosted regression trees to combine traditional linguistic features and neural network-based matching features.
(3) ECUN	See Table 5.4.
(4) LSTM	A classic architecture of recurrent neural network. In this research, it is used to concatenate the question subject and question body as an entirety, to obtain question representation and answer representation.
(5) LSTM-subject-body	A LSTM-based model. It is applied on question subject and question body respectively, and the results are concatenated to obtain the final representation of question.
(6) QCN	See Table 6.6.
(9) Himu-QAAN (Our model)	

Table 4.7: Baseline models of the SemEval2017 dataset.

Model	**F1**	**Acc**
(1) JAIST	0.7896	0.7910
(2) HITSZ-ICRC	0.7652	0.7611
(3) Graph-cut	0.8055	0.7980
(4) FCCRF	0.8150	0.8050
(5) BGMN	0.7723	0.7840
(6) CNN-LSTM-CRF	0.8222	0.8224
(7) QCN	0.8391	0.8224
(8) Himu-QAAN (our)	**0.8462**	**0.8365**

Table 4.8: Comparisons on the SemEval 2015 dataset.

Model	**MAP**	**F1**	**Acc**
(1) KeLP	0.8843	0.6987	0.7389
(2) Beihang-MSRA	0.8824	0.6840	0.5198
(3) ECNU	0.8672	0.7767	0.7843
(4) LSTM	0.8632	0.7441	0.7569
(5) LSTM-subject-body	0.8711	0.7450	0.7728
(6) QCN	0.8851	0.7811	0.8071
(7) Himu-QAAN (our)	**0.8962**	**0.7915**	**0.8178**

Table 4.9: Comparisons on the SemEval 2017 dataset.

Model	**MAP**	**F1**	**Acc**
(1) JAIST	0.7685	0.6856	0.7058
(2) HITSZ-ICRC	0.7384	0.6573	0.6785
(3) Graph-cut	0.7695	0.6879	0.7136
(4) FCCRF	0.7826	0.7052	0.7286
(5) BGMN	0.7489	0.6645	0.6827
(6) ECUN	0.7568	0.6758	0.6942
(7) CNN-LSTM-CRF	0.7926	0.7135	0.7489
(7) Himu-QAAN (our)	**0.8368**	**0.7394**	**0.7865**

Table 4.10: Comparisons on the Yahoo! Answers Dataset.

Model	Reference and Description
(1) w/o task-specific word embeddings	300-dimensional GloVe word vectors trained on Wikipedia 2014 and Gigaword 5.
(2) w/o character embeddings	600-dimensional GloVe word vectors trained on the domain-specific unannotated corpus
(3) without cross-attention	uses an inner attention over question words, and a hierarchical inner attention over answer words and sentences.
(4) Cross Attention between subject-body concatenation and answer	concatenates Question-subject and question-body and uses the entirety as input.
(5) Only Cross Attention between subject and answer	Only takes question-subject as input, and then uses question-subject and answer for cross attention.
(6) Only Cross Attention between body and answer	Only takes question-body as input, and then uses question-body and answer for cross attention.

Table 4.11: Experimental Methods and Description for comparisons on SemEval 2017 dataset.

Model	**Acc**
(1) w/o task-specific word embeddings	0.8085
(2) w/o character embeddings	0.7978
(3) without cross-attention	0.7659
(4) Cross Attention between subject-body concatenation and answer	0.7858
(5) Only Cross Attention between subject and answer	0.7769
(6) Only Cross Attention between body and answer	0.8072
(7) Himu-QAAN (ours)	0.8178

Table 4.12: Ablation studies on the SemEval 2017 dataset.

Model	Reference and Description
(1) w/o task-specific word embeddings	word embeddings are initialized with the 300-dimensional GloVe word vectors trained on Wikipedia 2014 and Gigaword 5.
(2) w/o character embeddings	word level embeddings are only composed of 600-dimensional GloVe word vectors trained on the domain-specific unannotated corpus
(3) without cross-attention	To evaluate the effect of using cross-attention mechanism on model performance, we evaluate the performance of a variant of HASAN that does not use cross attention between question and answer words. This model uses an inner attention over question words, and a hierarchical inner attention over answer words and sentences.
(4) without word inter attention	not use word inter attention.
(5) without sentence inter attention	not use sentence inter attention.

Table 4.13: Experimental Methods and Description for comparisons on Yahoo! Answers dataset.

Model	**MAP**	**F1**	**Acc**
(1) w/o task-specific word embeddings	0.8287	0.7287	0.7768
(2) w/o character embeddings	0.8139	0.7065	0.7649
(3) without cross-attention	0.7976	0.6859	0.7476
(4) without word inter attention	0.8025	0.6948	0.7542
(5) without sentence inter attention	0.8086	0.7012	0.7638
(6) Himu-QAAN (our model)	0.8368	0.7394	0.7865

Table 4.14: Ablation studies on the Yahoo! Answers dataset.

5 Duplicate Question Detection Task

5.1 Task Description

In this research, the task of duplicate question detection in CQA can be denoted as a tuple of five elements $(Q1, A1, Q2, A2, y)$:

$Q1 = [q1^1, q1^2, \cdots, q1^m]$ represents question 1 whose length is m;

$Q2 = [q2^1, q2^2, \cdots, q2^e]$ represents question 2 whose length is e;

$A1 = [a1^1, a1^2, \cdots, a1^n]$ is the paired answer of question 1, whose length is n;

$A2 = [a2^1, a2^2, \cdots, a2^f]$ denotes the paired answer of question 2, whose length is f;

$y \in Y$ represents whether the two archived questions (Q1 and Q2) are semantically equivalent or not. $Y = \{Good, Bad\}$, where "*Good*" means Q1 and Q2 are semantically equivalent; "*Bad*" means Q1 and Q2 are not.

Generally, the task of duplicate question detection can be condensed as follows: given a set of {Q1, A1, Q2, A2}, our model Bert-QAnet assigns a label to each answer, based on the conditional probability, i.e., $Pr(yQ1, A1, Q2, A2)$.

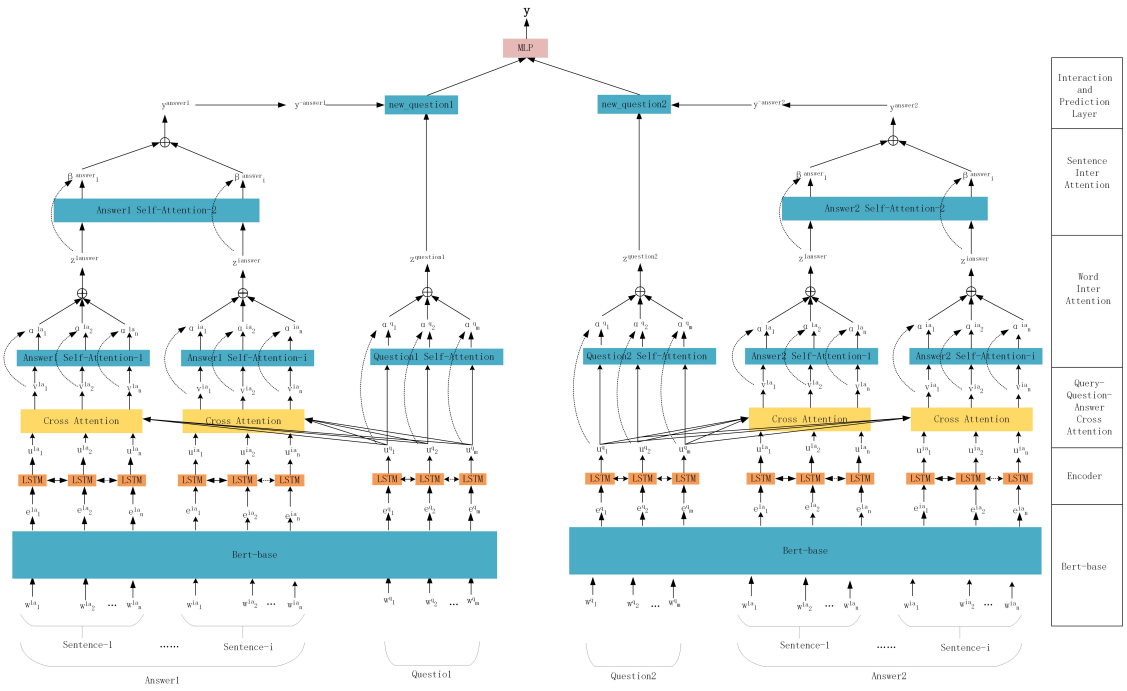

Figure 5.1: Overview of Our Proposed Model Bert-QAnet.

5.2 Overview of The Proposed Model

The structure of Bert-QAnet consists of six layers, including BERT Encoder, Cross-Attention, Word Inter Attention, Sentence Inter Attention and Classifier. These networks are assembled layer by layer from bottom to top. The flow-chart of our proposed framework is demonstrated in Figure 5.1. The same processing operation is performed for two archived Q-A pairs (Q1-A1 and Q2-A2), which guarantees the symmetry of our model.

5.3 BERT Encoder

BERT model, proposed by Devlin et al.[**?**], is used as the encoder layer to convert question subjects, question bodies, and answers into H-dimensional encoded forms containing different contextual representations. We define $\{w_t^{subject}\}_{t=1}^{m}$, $\{w_t^{body}\}_{t=1}^{g}$ and $\{w_t^{answer}\}_{t=1}^{n}$ as word sets of candidate question subjects, candidate question bodies, and candidate answers. Respectively, m, g and n are the length of each question subject, each question body, and each answer. We obtain $U^{subject}$, u_t^{body}, $U^{answer1}$ and $U^{answer2}$ by encoding question subjects $\{w_t^{subject}\}_{t=1}^{m}$ and answers $\{w_t^{answer}\}_{t=1}^{n}$. Also, we use BERT model to encode bodies $\{w_t^{body}\}_{t=1}^{g}$ and answers $\{w_t^{answer}\}_{t=1}^{n}$ into $u_t^{subject}$, u_t^{body}, $u_t^{answer1}$ and $u_t^{answer2}$.

The related equations of this layer are displayed as follows: $\mathrm{u}_t^{subject}, u_t^{answer1} = BERT_{sub-ans}([w_t^{subject}; w_t^{answer}])$

$u_t^{body}, u_t^{answer2} = BERT_{body-ans}([w_t^{body}; w_t^{answer}])$

$U^{suject} = \{u_t^{subject}\}_{t=1}^{m} \in R^{m \times H}$

$U^{body} = \{u_t^{body}\}_{t=1}^{g} \in R^{g \times H}$

$U^{answer1} = \{u_t^{answer1}\}_{t=1}^{n} \in R^{n \times H}$

$U^{answer2} = \{u_t^{answer2}\}_{t=1}^{n} \in R^{n \times H}$

Other layers are similar to Himu-QAAN model, hence, these details are not described again.

Yahoo! Answers Corpus			
Statistics	Train	Dev	Test
Number of questions	3500	500	1000
Number of answers	3500	500	1000
Average length of question	7.28	7.05	7.16
Average length of answer	38.65	36.96	37.54
Stack Overflow Corpus			
Statistics	Train	Dev	Test
Number of questions	3500	500	1000
Number of answers	3500	500	1000
Average length of question	7.89	7.43	7.65
Average length of answer	39.54	38.92	39.32

Table 5.1: Statistics of the two datasets.

5.4 Datasets

To train and evaluate our model, we used two corpora: the Yahoo! Answers dataset and the Stack Overflow dataset. The statistics of the two corpora are shown in Table 5.1.

5.5 Settings and Hyper-parameters

For the text pre-processing procedure, we applied the NLTK toolkit, including capitalization conversion, stemming, removal of stop words etc., to each Q-A pair.

The algorithm we chose for optimization is the Adam Optimizer, with momentum coefficient 0.9, and the second momentum coefficient 0.999. We selected the parameters with the best performance on validation set, and then evaluated the performance of our model on test sets. The hyper-parameters are listed in Table 5.2.

5.6 Results and Analyses

Two evaluation metrics are adpoted, *F1* and *Acc* (accuracy) to compare the performance of model Bert-QAnet with nine baseline models (1) to (10) as shown in Table 5.3. Specifically, (1) and (2) are both sentence encoding-based models; (3) to (9) are models that use various cross sentence features; (10) is the most advanced pre-trained model. On the dataset, the model Bert-QAnet used the Q-A pairs information, while other models were solely trained on the paired question dataset. First, comparison experiments are conducted on the Yahoo! Answers dataset. The results are shown in Table 5.4. Our model achieved 0.8723 in terms of *F1* and 0.8967 in terms of *Acc*, outperforming the ten baselines.

Hyper-parameters	Value
LSTM hidden size	300
Batch size	128
Learning rate	0.001
L2 regularization parameters	0.001
Gradient Clipping	5
Early stop patience	10
LSTM dropout rate	0.5
BERT hidden size	768
BERT hidden layers	12

Table 5.2: Settings and hyper-parameters.

In addition, Bert-QAnet and five strong baselines (BiMPM, pt-DECATT, ESIM, AF-DMN and DIIN) are compared on the Stack Overflow dataset, as demonstrated in Table 5.5. This method achieved state-of-the-art performance on all evaluation metrics. Since our model was the only one that used paired answer information, it demonstrated that *answer information* could be an vitally important knowledge source for duplicate question detection.

5.7 Answer Information Research

To further verify the validity of answer information for the duplicate question detection task, we implemented additional experiments on the Yahoo! Answers dataset.

From Table 5.6, it can be observed that using only answer information can also achieve good results (Row 1). Specifically, we trained the DIIN method on the paired answer dataset, and the trained model achieved 0.8024 in terms of Acc on the test set. This is because *answer* usually explains *question* in detail, providing sufficient reinforcement for duplication detection. Moreover, it is easy to notice that concatenation might introduce additional noise which may weaken the final performance. To verify this notion, we compared model (2) with (3), and experimental results were consistent with our inferences. This confirms that our model breaks through the bottleneck described in our Introduction.

Overall, model Bert-QAnet is a strong exemplar of applying paired answer for duplication detection, while appropriately avoiding additional noise and redundancies. This model achieves state-of-the-art performance and realizes a breakthrough of this task.

5.8 Ablation Study

An ablation study is conducted on the base model to examine the objective of each component on the Yahoo! Answers dataset. The results are presented in Table 5.7.

First, BERT embedding and GloVe word representations (proposed by [**?**]) is compared. GloVe is one of the most famous examples of distributed representations for text, which is regarded as one of the key breakthroughs for deep learning techniques in challenging natural language processing tasks. Especially, when trained on a domain-specific corpus, features of words are captured more accurately. Specifically, in our work, we trained the dataset with GloVe to obtain 300-dimensional initialized word vectors. The representations of Out-of-Vocabulary (OOV) are set to zero. The comparisons show that *Acc* dropped to 0.8967 when we replaced BERT embedding with GloVe. This proves that the BERT model advances the embedding of text to a more significant level.

Next, the contribution of cross-attention to this model is studied. After excluding cross-attention, the performance of our model dropped to 0.9042 on the test set. This comparison proves that cross-attention can capture the correlation between two sentences, which is crucial for this task.

Thirdly, when the word inner attention was excluded, detection performance on the test set dropped to 0.9182. The main reason for this is word inner attention not

only simulated the temporal interaction of words, but also processed the long-term dependencies in long sentences to obtain a stronger semantic representation.

And finally, when the sentence inner attention was excluded, the performance of our model dropped to 0.8769, proving that sentence inner attention can obtain the global semantic information, which is a powerful supplement to word inner attention.

Therefore, due to the effective combining of these various methods, our model integrated valuable features from paired answers for duplication detection and successfully filtered out noise introduced by answers.

Model	Description
(1) InferSent [12]	a sentence encoding-based model.
(2) SSE [56]	a shortcut-stacked sequential sentence encoder for multi-domain natural language inference (NLI).
(3) PWIM [22]	a hybrid of ConvNet and Bi-LSTMs for the semantic textual similarity measurement task.
(4) pt-DECATT[75]	a decomposable attention model for question paraphrase identification (QPI).
(5) ESIM [9]	an enhanced LSTM model for NLI task.
(6) DIIN [58]	a high-level understanding of the sentence pair.
(7) AF-DMN [17]	an attention-fused deep matching network.
(8) Multi-Perspective-CNN [84]	uses multi-perspective cosine matching for the paraphrase identification task.
(9) BiMPM [84]	a bilateral multi-perspective matching model under the "matching-aggregation" framework.
(10) Bert-base [86]	the most famous pre-training model, which can be fine-tuned with just one additional output layer.
(11) BERT-QAnet	**Our model**

Table 5.3: Experimental Methods and Descriptions.

Model	F1	Acc
(1) InferSent	0.8012	0.8369
(2) SSE	0.8236	0.8537
(3) PWIM	0.7367	0.7625
(4) Multi-Perspective-CNN	0.7698	0.8065
(5) BiMPM	0.8372	0.8676
(6) pt-DECATT	0.8198	0.8516
(7) ESIM	0.8027	0.8432
(8) AF-DMN	0.8412	0.8695
(9) DIIN	0.8526	0.8782
(10) BERT-base	0.8769	0.9136
(11) Bert-QAnet (ours)	**0.8895**	**0.9264**

Table 5.4: Experimental results on Yahoo! Answer dataset.

Model	F1	Acc
(1) BiMPM	0.8165	0.8471
(2) pt-DECATT	0.7836	0.8296
(3) ESIM	0.7652	0.7967
(4) AF-DMN	0.8287	0.8506
(5) DIIN	0.8391	0.8634
(6) Bert-QAnet (ours)	**0.8558**	**0.8792**

Table 5.5: Experimental results on Stack Overflow dataset.

Model	F1	Acc
(1) DIIN (Answer Pairs Only)	0.7562	0.8024
(2) DIIN (Q-A Pairs)	0.8076	0.8385
(3) DIIN (Question Pairs Only)	0.8526	0.8782
(4) Bert-QAnet (ours)	**0.8723**	**0.8967**

Table 5.6: Experimental results of answer information research.

Model	Acc
(1) Bert-QAnet (replaced BERT with GloVe)	0.8967
(2) Bert-QAnet (without cross-attention)	0.9042
(3) Bert-QAnet (without word inner attention)	0.9182
(4) Bert-QAnet (without sentence inner attention)	0.9205
(5) Bert-QAnet (ours)	0.9264

Table 5.7: Ablation study.

6 Conclusion

Community question answering (CQA) systems are gaining popularity online. One can freely ask any question and expect some good, honest answers, but it takes efforts and much time to go through all possible answers and winnow the most relevant one attention networks for answer selection, which take the relationship between questions and answers as important information. For investigating the role of answers' information corresponding to the questions, QAAN studies the correlation between question and paired answer, taking question as the primary part of the question representation, and the answer information is aggregated based on similarity and disparity with the answer. To effectively answer both factoid and non-factoid questions with various length, model Himu-QAAN applies deep attention mechanism at word, sentence, and document level, utilizing characteristics of linguistic knowledge to explore the complex relationship among question subjects, question bodies and answers. Cross-attention mechanism is used between question subjects and answers, question bodies and answers to capture interactive features.

Moreover, inner attention on question subjects as well as hierarchical inner attention on question bodies and answers help to assign different weights to each word so as to determine important words in a sentence and important sentences in a document. Through integrating attention-question-subjects, attention-question-bodies and attention-answers, Himu-QAAN model gets final results which achieve significant performance outperforming all current answer selection models. In the future, our research group will mainly focus on improving the computing speed of Himu-QAAN to further level up the performance of our solution. Bert-QAnet applied BERT to encode text and extract text features. Model Bert-QAnet applied deep attention mechanism at word, sentence, and document level, respectively. Experiment results demonstrated that Bert-QAnet comprehensively used characteristics of linguistic knowledge to explore the complexity of different components. Bert-QAnet outperformed all baseline models, achieving state-of-the-art performance in duplicate question detection.

In the future, I would like to test the proposed framework on more real-world datasets. Also, I would like to explore the possibility of making the proposed models less complex.

Bibliography

[1] Ashutosh Adhikari, Achyudh Ram, Raphael Tang, and Jimmy J. Lin. Rethinking complex neural network architectures for document classification. In *NAACL-HLT*, 2019.

[2] João António Rodrigues, Chakaveh Saedi, Vladislav Maraev, João Silva, and António Branco. Ways of asking and replying in duplicate question detection. In *Proceedings of the 6th Joint Conference on Lexical and Computational Semantics (*SEM 2017)*, pages 262–270, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[3] Kyoungman Bae and Youngjoong Ko. Efficient question classification and retrieval using category information and word embedding on cqa services. *Journal of Intelligent Information Systems*, pages 1–23, 2019.

[4] Alberto Barrón-Cedeño, Simone Filice, Giovanni Da San Martino, Shafiq Joty, Lluís Màrquez, Preslav Nakov, and Alessandro Moschitti. Thread-level information for comment classification in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 687–693, Beijing, China, July 2015. Association for Computational Linguistics.

[5] Adam Berger, Rich Caruana, David Cohn, Dayne Freitag, and Vibhu Mittal. Bridging the lexical chasm: Statistical approaches to answer-finding. In *Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '00, page 192–199, New York, NY, USA, 2000. Association for Computing Machinery.

[6] Robin D. Burke, Kristian J. Hammond, Vladimir A. Kulyukin, Steven L. Lytinen, N. Tomuro, and S. Schoenberg. Question answering from frequently asked question files: Experiences with the faq finder system. Technical report, USA, 1997.

[7] Long Chen, Dell Zhang, and Mark Levene. Question retrieval with user intent. SIGIR '13, page 973–976, New York, NY, USA, 2013. Association for Computing Machinery.

[8] Long Chen, Dell Zhang, and Levene Mark. Understanding user intent in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 823–828, New York, NY, USA, 2012. Association for Computing Machinery.

[9] Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. Enhanced LSTM for natural language inference. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1657–1668, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[10] Zheqian Chen, Chi Zhang, Zhou Zhao, Chengwei Yao, and Deng Cai. Question retrieval for community-based question answering via heterogeneous social influential network. *Neurocomputing*, 285:117–124, 2018.

[11] Michael Collins and Nigel Duffy. Convolution kernels for natural language. In *Proceedings of the 14th International Conference on Neural Information Processing Systems: Natural and Synthetic*, NIPS'01, page 625–632, Cambridge, MA, USA, 2001. MIT Press.

[12] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[13] Min-Yuh Day and Yu-Ling Kuo. A study of deep learning for factoid question answering system. *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 419–424, 2020.

[14] Yang Deng, Wenxuan Zhang, and Wai Lam. *Opinion-Aware Answer Generation for Review-Driven Question Answering in E-Commerce*, page 255–264. Association for Computing Machinery, New York, NY, USA, 2020.

[15] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In

Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.

[16] Cícero dos Santos, Luciano Barbosa, Dasha Bogdanova, and Bianca Zadrozny. Learning hybrid representations to retrieve semantically equivalent questions. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 694–699, Beijing, China, July 2015. Association for Computational Linguistics.

[17] Chaoqun Duan, Lei Cui, Xinchi Chen, Furu Wei, Conghui Zhu, and Tiejun Zhao. Attention-fused deep matching network for natural language inference. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 4033–4040. International Joint Conferences on Artificial Intelligence Organization, 7 2018.

[18] Wenzheng Feng, Yuehua Wu, Wei Wu, Zhoujun Li, and M. Zhou. Beihang-msra at semeval-2017 task 3: A ranking system with neural matching features for community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL)*, page 280–286, 2017.

[19] Simone Filice, Danilo Croce, Alessandro Moschitti, and Roberto Basili. KeLP at SemEval-2016 task 3: Learning semantic relations between questions and answers. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 1116–1123, San Diego, California, June 2016. Association for Computational Linguistics.

[20] Simone Filice, Giovanni Da San Martino, and Alessandro Moschitti. Kelp at semeval-2017 task 3: Learning pairwise patterns in community question answering. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL)*, page 326–333, 2017.

[21] Marc Franco-Salvador, Sudipta Kar, Thamar Solorio, and Paolo Rosso. UH-PRHLT at SemEval-2016 task 3: Combining lexical and semantic-based features for community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 814–821, San Diego, California, June 2016. Association for Computational Linguistics.

[22] Hua He and Jimmy J. Lin. Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the*

2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (HLT-NAACL), pages 937–948, 2016.

[23] Michael Heilman and Noah A. Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1011–1019, Los Angeles, California, June 2010. Association for Computational Linguistics.

[24] D. Hoogeveen, Andrew Bennett, Yitong Li, Karin M. Verspoor, and Timothy Baldwin. Detecting misflagged duplicate questions in community question-answering archives. In *Proceedings of the 12th International AAAI conference on Web and Social Media (ICWSM)*, pages 112–120, 2018.

[25] Yongshuai Hou, C. Tan, Xiaolong Wang, Yaoyun Zhang, Jun Xu, and Q. Chen. Hitsz-icrc: Exploiting classification approach for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval@NAACL-HLT)*, page 196–202, 2015.

[26] Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. Convolutional neural network architectures for matching natural language sentences. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 27. Curran Associates, Inc., 2014.

[27] Haifeng Hu, B. Liu, Baoxun Wang, Ming Liu, and Xiaolong Wang. Multimodal dbn for predicting high-quality answers in cqa portals. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 843–847, 2013.

[28] Ze Hu, Zhan Zhang, Haiqin Yang, Qing Chen, Rong Zhu, and Decheng Zuo. Predicting the quality of online health expert question-answering services with temporal features in a deep learning framework. *Neurocomputing*, 275:2769–2782, 2018.

[29] Weiyi Huang, Qiang Qu, and Min Yang. Interactive knowledge-enhanced attention network for answer selection. *Neural Computing and Applications*, pages 1–17, 2020.

[30] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding semantically similar questions based on their answers. In *Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '05, page 617–618, New York, NY, USA, 2005. Association for Computing Machinery.

[31] Jiwoon Jeon, W. Bruce Croft, and Joon Ho Lee. Finding similar questions in large question and answer archives. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, page 84–90, New York, NY, USA, 2005. Association for Computing Machinery.

[32] Valentin Jijkoun and Maarten de Rijke. Retrieving answers from frequently asked questions pages on the web. In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management*, CIKM '05, page 76–83, New York, NY, USA, 2005. Association for Computing Machinery.

[33] Shafiq R. Joty, A. Barrón-Cedeño, Giovanni Da San Martino, Simone Filice, Lluís Màrquez i Villodre, Alessandro Moschitti, and Preslav Nakov. Global thread-level inference for comment classification in community question answering. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 573–578, 2015.

[34] Shafiq R. Joty, Lluís Màrquez i Villodre, and Preslav Nakov. Joint learning with global inference for comment classification in community question answering. In *HLT-NAACL*, 2016.

[35] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 655–665, Baltimore, Maryland, June 2014. Association for Computational Linguistics.

[36] Yoon Kim. Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1746–1751, Doha, Qatar, October 2014. Association for Computational Linguistics.

[37] Yoon Kim, Yacine Jernite, David Sontag, and Alexander M. Rush. Character-aware neural language models. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2741–2749. AAAI Press, 2016.

[38] Alexandros Komninos and S. Manandhar. Dependency based embeddings for sentence classification tasks. In *Proceedings of the 2016 conference of the North American chapter of the Association for Computational Linguistics: human language technologies (NAACL HLT)*, pages 1490–1500, 2016.

[39] Md Tahmid Rahman Laskar, Xiangji Huang, and Enamul Hoque. Contextualized embeddings based transformer encoder for sentence similarity modeling in answer selection task. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC)*, 2020.

[40] Phong Le and Willem H. Zuidema. The forest convolutional network: Compositional distributional semantics with a neural chart and without binarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1155–1164, 2015.

[41] Bengio Y. Hinton G LeCun, Y. Deep learning. *Nature*, 521:436–444, 2015.

[42] Y. LeCun. Deep learning & convolutional networks. *2015 IEEE Hot Chips 27 Symposium (HCS)*, pages 1–95, 2015.

[43] Y. LeCun. The power and limits of deep learning. *Research-Technology Management*, 61:22 – 27, 2018.

[44] Tao Lei, Hrishikesh Joshi, R. Barzilay, T. Jaakkola, K. Tymoshenko, Alessandro Moschitti, and Lluís Màrquez i Villodre. Semi-supervised question retrieval with gated convolutions. In *Proceedings of the North American Chapter of the Association for Computational Linguistics (NAACL 2016)*, 2016.

[45] Zhouhan Lin, Minwei Feng, C. D. Santos, Mo Yu, Bing Xiang, Bowen Zhou, and Yoshua Bengio. A structured self-attentive sentence embedding. abs/1703.03130, 2017.

[46] Zachary Chase Lipton and J. Steinhardt. Troubling trends in machine learning scholarship. *Queue*, 17:45 – 77, 2019.

[47] Yoshitomo Matsubara, Thuy Vu, and Alessandro Moschitti. Reranking for efficient transformer-based answer selection. *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2020.

[48] Stephen Merity, N. Keskar, and R. Socher. Regularizing and optimizing lstm language models. *ArXiv*, abs/1708.02182, 2018.

[49] Donald Metzler and W. Croft. Analysis of statistical question classification for fact-based questions. *Information Retrieval*, 8:481–504, 2005.

[50] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems*, volume 26, pages 3111–3119. Curran Associates, Inc., 2013.

[51] Salman Mohammed, Peng Shi, and Jimmy Lin. Strong baselines for simple question answering over knowledge graphs with and without neural networks. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 291–296, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[52] Nafise Sadat Moosavi and Michael Strube. Lexical features in coreference resolution: To be used with caution. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 14–19, Vancouver, Canada, July 2017. Association for Computational Linguistics.

[53] Lili Mou, Hao Peng, Ge Li, Yan Xu, L. Zhang, and Zhi Jin. Discriminative neural sentence modeling by tree-based convolution. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2315–2325, 2015.

[54] Vanessa Murdock and W. Croft. Simple translation models for sentence retrieval in factoid question answering. In *Proceedings of the Information Retrieval for Question Answering Workshop at SIGIR*, pages 31–35, 2004.

[55] Preslav Nakov, Lluís Màrquez, Alessandro Moschitti, Walid Magdy, Hamdy Mubarak, Abed Alhakim Freihat, Jim Glass, and Bilal Randeree. SemEval-2016 task 3: Community question answering. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016)*, pages 525–545, San Diego, California, June 2016. Association for Computational Linguistics.

[56] Yixin Nie and Mohit Bansal. Shortcut-stacked sentence encoders for multi-domain inference. abs/1708.02312:41–45, 2017.

[57] Bian Ning, Xianpei Han, B. Chen, and Le Sun. Benchmarking knowledge-enhanced commonsense question answering via knowledge-to-text transformation. *ArXiv*, abs/2101.00760, 2021.

[58] Nikolaos Pappas and Andrei Popescu-Belis. Multilingual hierarchical attention networks for document classification. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1015–1025, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.

[59] Jeffrey Pennington, Richard Socher, and Christopher Manning. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar, October 2014. Association for Computational Linguistics.

[60] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana, June 2018. Association for Computational Linguistics.

[61] Xipeng Qiu and Xuanjing Huang. Convolutional neural tensor network architecture for community-based question answering. In *Proceedings of the 24th International Conference on Artificial Intelligence*, IJCAI'15, page 1305–1311. AAAI Press, 2015.

[62] Fatemeh Riahi, Zainab Zolaktaf, Mahdi Shafiei, and Evangelos Milios. Finding expert users in community question answering. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12 Companion, page 791–798, New York, NY, USA, 2012. Association for Computing Machinery.

[63] Stefan Riezler, Alexander Vasserman, Ioannis Tsochantaridis, Vibhu Mittal, and Yi Liu. Statistical machine translation for query expansion in answer retrieval. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 464–471, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[64] Dan Roth. Learning to resolve natural language ambiguities: A unified approach. National Conference on Artificial Intelligence (AAAI), page 806–813, USA, 1998. American Association for Artificial Intelligence.

[65] D. Sculley, Jasper Snoek, Alexander B. Wiltschko, and A. Rahimi. Winner's curse? on pace, progress, and empirical rigor. In *ICLR*, 2018.

[66] Minjoon Seo, Aniruddha Kembhavi, Ali Farhadi, and Hannaneh Hajishirzi. Bidirectional attention flow for machine comprehension. abs/1611.01603, 2017.

[67] Faiz Ali Shah, Kairit Sirts, and Dietmar Pfahl. Simple app review classification with only lexical features. In *Proceedings of the International Conference on Software Technologies (ICSOFT)*, 2018.

[68] Tao Shen, Tianyi Zhou, Guodong Long, Jing Jiang, Shirui Pan, and C. Zhang. Disan: Directional self-attention network for rnn/cnn-free language understanding. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 5446–5455, 2018.

[69] Yikang Shen, Wenge Rong, Nan Jiang, Baolin Peng, Jie Tang, and Z. Xiong. Word embedding based correlation model for question/answer matching. In *Proceedings of the 31st AAAI Conference on Artificial Intelligence (AAAI)*, page 3511–3517, 2017.

[70] Anna Shtok, Gideon Dror, Yoelle Maarek, and Idan Szpektor. Learning from the past: Answering new questions with past answers. In *Proceedings of the 21st International Conference on World Wide Web*, WWW '12, page 759–768. Association for Computing Machinery, 2012.

[71] Amit Singh and Karthik Visweswariah. Cqc: Classifying questions in cqa websites. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, CIKM '11, page 2033–2036, New York, NY, USA, 2011. Association for Computing Machinery.

[72] Ivan Srba and M. Bieliková. A comprehensive survey and classification of approaches for community question answering. *ACM Transactions on the Web (TWEB)*, 10:1 – 63, 2016.

[73] Dongge Tang, Wenge Rong, Shuang Qin, Jianxin Yang, and Z. Xiong. A n-gated recurrent unit with review for answer selection. *Neurocomputing*, 371:158–165, 2020.

[74] Yi Tay, Anh Tuan Luu, and S. C. Hui. Cross temporal recurrent networks for ranking question answer pairs. In *The Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 5512–5519, 2018.

[75] Gaurav Singh Tomar, Thyago Duque, Oscar Täckström, Jakob Uszkoreit, and Dipanjan Das. Neural paraphrase identification of questions with noisy

pretraining. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 142–147, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.

[76] Quan Hung Tran, Vu Duc Tran, Tu Thanh Vu, Minh Le Nguyen, and Son Bao Pham. JAIST: Combining multiple features for answer selection in community question answering. In *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*, pages 215–219, Denver, Colorado, June 2015. Association for Computational Linguistics.

[77] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, undefinedukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA, 2017. Curran Associates Inc.

[78] Shengxian Wan, Yanyan Lan, Jiafeng Guo, Jun Xu, Liang Pang, and Xueqi Cheng. A deep architecture for semantic matching with multiple positional sentence representations. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 2835–2841. AAAI Press, 2016.

[79] Baoxun Wang, B. Liu, Chengjie Sun, Xiaolong Wang, and Lin Sun. Extracting chinese question-answer pairs from online forums. *2009 IEEE International Conference on Systems, Man and Cybernetics*, pages 1159–1164, 2009.

[80] Baoxun Wang, Xiaolong Wang, Chengjie Sun, Bingquan Liu, and Lin Sun. Modeling semantic relevance for question-answer pairs in web social communities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1230–1238, Uppsala, Sweden, July 2010. Association for Computational Linguistics.

[81] Kai Wang, Zhaoyan Ming, and Tat-Seng Chua. A syntactic tree matching approach to finding similar questions in community-based qa services. In *Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '09, page 187–194, New York, NY, USA, 2009. Association for Computing Machinery.

[82] Mengqiu Wang, Noah A. Smith, and Teruko Mitamura. What is the Jeopardy model? a quasi-synchronous grammar for QA. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages

22–32, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

[83] Xianzhi Wang, Chaoran Huang, Lina Yao, B. Benatallah, and Manqing Dong. A survey on expert recommendation in community question answering. *Journal of Computer Science and Technology*, 33:625–653, 2018.

[84] Zhiguo Wang, Wael Hamza, and Radu Florian. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 4144–4150, 2017.

[85] Zhiguo Wang, Haitao Mi, and Abraham Ittycheriah. Sentence similarity learning by lexical decomposition and composition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1340–1349, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.

[86] Dirk Weissenborn, Georg Wiese, and Laura Seiffe. Making neural QA as simple as possible but not simpler. In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*, pages 271–280, Vancouver, Canada, August 2017. Association for Computational Linguistics.

[87] GuoShun Wu, Yixuan Sheng, Man Lan, and Yuanbin Wu. Ecnu at semeval-2017 task 3: Using traditional and deep learning methods to address community question answering task. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval@ACL)*, page 365–369, 2017.

[88] Haocheng Wu, Wei Wu, Ming Zhou, Enhong Chen, Lei Duan, and Heung-Yeung Shum. Improving search relevance for short queries in community question answering. In *Proceedings of the 7th ACM International Conference on Web Search and Data Mining*, WSDM '14, page 43–52, New York, NY, USA, 2014. Association for Computing Machinery.

[89] Wei Wu, Xu Sun, and Houfeng Wang. Question condensing networks for answer selection in community question answering. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1746–1755, Melbourne, Australia, July 2018. Association for Computational Linguistics.

[90] Wei Wu, Houfeng Wang, and Sujian Li. Bi-directional gated memory networks for answer selection. In *Chinese Computational Linguistics and Natural*

Language Processing Based on Naturally Annotated Big Data (CCL), page 251–262, 2017.

[91] Yan Wu, Qi Zhang, and Xuanjing Huang. Efficient near-duplicate detection for q&a forum. In *Proceedings of 5th International Joint Conference on Natural Language Processing (IJCNLP)*, pages 1001–1009, 2011.

[92] Yang Xiang, Xiaoqiang Zhou, Q. Chen, Zhihui Zheng, Buzhou Tang, Xiaolong Wang, and Yang Qin. Incorporating label dependency for answer quality tagging in community question answering via cnn-lstm-crf. In *Proceedings of the 26th International Conference on Computational Linguistics, Proceedings of the Conference: Technical Papers (COLING)*, page 1231–1241, 2016.

[93] Xiaobing Xue, Jiwoon Jeon, and W. Bruce Croft. Retrieval models for question and answer archives. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, page 475–482, New York, NY, USA, 2008. Association for Computing Machinery.

[94] Haitian Yang, Weiqing Huang, Xuan Zhao, Yan Wang, Yuyan Chen, Bin Lv, Rui Mao, and Ning Li. Amqan: Adaptive multi-attention question-answer networks for answer selection. In *Proceedings of the 2000 European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML/PKDD)*, 2020.

[95] Pengcheng Yang, Xu Sun, Wei Li, Shuming Ma, Wei Wu, and Houfeng Wang. SGM: Sequence generation model for multi-label classification. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 3915–3926, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[96] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, San Diego, California, June 2016. Association for Computational Linguistics.

[97] Xuchen Yao, Benjamin Van Durme, Chris Callison-Burch, and Peter Clark. Answer extraction as sequence tagging with tree edit distance. In *HLT-NAACL*, 2013.

[98] Seunghyun Yoon, Franck Dernoncourt, Doo Soon Kim, Trung Bui, and Kyomin Jung. A compare-aggregate model with latent clustering for answer selection. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, 2019.

[99] Adams Wei Yu, David Dohan, Minh-Thang Luong, Rui Zhao, Kai Chen, Mohammad Norouzi, and Quoc V. Le. Qanet: Combining local convolution with global self-attention for reading comprehension. abs/1804.09541, 2018.

[100] Puxuan Yu, Zhiqi Huang, Razieh Rahimi, and James Allan. Corpus-based set expansion with lexical features and distributed representations. *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2019.

[101] Phil Blunsom Yu Lei, Karl Moritz Hermann and Stephen Pulman. Deep learning for answer sentence selection. *ArXiv*, abs/1412.1632, 2014.

[102] Sha Yuan, Y. Zhang, Jie Tang, Wendy Hall, and J. Cabota. Expert finding in community question answering: a review. *Artificial Intelligence Review*, 53:843–874, 2019.

[103] Wei Emma Zhang, Quan Z. Sheng, Jey Han Lau, and Ermyas Abebe. Detecting duplicate posts in programming qa communities via latent semantics and association rules. In *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, page 1221–1229. International World Wide Web Conferences Steering Committee, 2017.

[104] Xiaodong Zhang, Sujian Li, Lei Sha, and Houfeng Wang. Attentive interactive neural networks for answer selection in community question answering. In Satinder P. Singh and Shaul Markovitch, editors, *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 3525–3531. AAAI Press, 2017.

[105] Xiaodong Zhang, Xu Sun, and Houfeng Wang. Duplicate question identification by integrating framenet with neural networks. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI)*, pages 6061–6068, 2018.

[106] Y. Zhang, D. Lo, Xin Xia, and Jianling Sun. Multi-factor duplicate question detection in stack overflow. *Journal of Computer Science and Technology*, 30:981–997, 2015.

[107] Guangyou Zhou, Li Cai, Jun Zhao, and Kang Liu. Phrase-based translation model for question retrieval in community question answer archives. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 653–662, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

[108] Xiaoqiang Zhou, Baotian Hu, Q. Chen, and Xiaolong Wang. Recurrent convolutional neural network for answer selection in community question answering. *Neurocomputing*, 274:8–18, 2018.

[109] Xiaoqiang Zhou, Baotian Hu, Qingcai Chen, Buzhou Tang, and Xiaolong Wang. Answer sequence learning with neural networks for answer selection in community question answering. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 713–718, Beijing, China, July 2015. Association for Computational Linguistics.

[110] Ming Zhu, Aman Ahuja, Wei Wei, and Chandan K. Reddy. A hierarchical attention retrieval model for healthcare question answering. In *The World Wide Web Conference*, WWW '19, page 2472–2482, New York, NY, USA, 2019. Association for Computing Machinery.