agronomy

MDPI

Article

ALBERT over Match-LSTM Network for Intelligent Questions Classification in Chinese

Xiaomin Wang [1], Haoriqin Wang [1,2,3], Guocheng Zhao [4], Zhichao Liu [1,3] and Huarui Wu [1,*]

[1] Beijing Research Center for Information Technology in Agriculture, Beijing 100097, China; wangxm007@nercita.org.cn (X.W.); wanghrq007@nercita.org.cn (H.W.); liuzc007@nercita.org.cn (Z.L.)
[2] College of Computer Science and Technology, Inner Mongolia University for Nationalities, Tongliao 028043, China
[3] School of Information and Electrical Engineering, Shenyang Agricultural University, Shenyang 110866, China
[4] School of Mechanical Electronic and Information Engineering, China University of Mining and Technology, Beijing 100083, China; xs206610@student.cumtb.edu.cn
* Correspondence: wuhr007@nercita.org.cn

Abstract: This paper introduces a series of experiments with an ALBERT over match-LSTM network on the top of pre-trained word vectors, for accurate classification of intelligent question answering and thus the guarantee of precise information service. To improve the performance of data classification, a short text classification method based on an ALBERT and match-LSTM model was proposed to overcome the limitations of the classification process, such as few vocabularies, sparse features, large amount of data, lots of noise and poor normalization. In the model, Jieba word segmentation tools and agricultural dictionary were selected to text segmentation, GloVe algorithm was then adopted to expand the text characteristic and weighted word vector according to the text of key vector, bi-directional gated recurrent unit was applied to catch the context feature information and multi-convolutional neural networks were finally established to gain local multidimensional characteristics of text. Batch normalization, Dropout, Global Average Pooling and Global Max Pooling were utilized to solve overfitting problem. The results showed that the model could classify questions accurately, with a precision of 96.8%. Compared with other classification models, such as multi-SVM model and CNN model, ALBERT+match-LSTM had obvious advantages in classification performance in intelligent Agri-tech information service.

Keywords: ALBERT; match-LSTM; natural language processing; classification; NQuAD

Citation: Wang, X.; Wang, H.; Zhao, G.; Liu, Z.; Wu, H. ALBERT over Match-LSTM Network for Intelligent Questions Classification in Chinese. *Agronomy* **2021**, *11*, 1530. https://doi.org/10.3390/agronomy11081530

Academic Editor: Andrea Sciarretta

Received: 28 June 2021
Accepted: 27 July 2021
Published: 30 July 2021

1. Introduction

It has been almost 60 years since the first successful knowledge-based question answering system for baseball was developed in 1963, but the intelligent QA machine for Chinese agriculture datasets is still Blue Ocean.

1.1. Goals of the Paper

Our intelligent question answering machine constrains answers to cloze-style reading comprehension datasets [1,2], also called domain-specific, different from the space of all possible spans. This work aims to improve the performance of question answering on the National Agricultural Technology and Education Cloud Platform (NJTG, http://njtg.nercita.org.cn/ (accessed 20 August 2017)) by applying classification technology. Our model is evaluated on data from NJTG, which is based on big data, cloud computation and mobile technology, with all kinds of agricultural technology educational resources—we call it the wing of Chinese agricultural technology. Agricultural administration departments at all levels, such as agricultural experts, agricultural technology-extension officers and farmers, who can access online education, online consultation, achievement promotion and marketing easily, will fasten the development of Chinese intelligent agriculture. Our

cloud platform integrates a "π" shape structure and serves different versions for different end-users: the national fundamental platform WEB version for agricultural management users, the NJTG app for all agricultural users and the intelligent farmer cloud app for farmers. It is funded by the China Agricultural Ministry and developed by the National Engineering Research Center for Information Technology in agriculture (abbr. NERCITA).

An intelligent Agricultural Question Answering Machine named "Xiaoman" is built based on our model, aiming to answer comprehension-style orchard worker's questions about various crops or fruits such as peaches given a sentence or passage. Xiaoman has three characteristics over other question answering systems. (1) Data source: we specifically concentrate on the NERCITA Question Answering Dataset (NQuAD), a large-scale agricultural dataset for reading comprehension and question answering which is collected through national agricultural technology and education cloud platform (NJTG); corresponding to Stanford Question Answering Dataset (SQuAD) [3]. (2) Question type: different questions are classified into six different types, particularly cloze-domain questions, which will be beneficial to the research community. (3) Scale: it holds more than 10 million question–answer pairs targeting agricultural technology, servicing orchard workers; it is the largest knowledge-based Chinese agricultural technology cloud platform so far. We are challenged with more realistic data sources, more types of questions and more scale, as illustrated in Table 2.

The release of the SQuAD advances reading comprehension and question answering (RC and QA) substantially. Cui et al. [4] can generate "attended-attention" automatically with a much simpler model AOA (attention over attention). Seo et al. [5] introduce bi-directional attention flow networks (BiDAF) to model question–passage pairs at multiple levels of granularity. A lively co-attention network attending the question and passage concurrently is presented by Xiong et al. [6,7], which can enhance answer predictions recurrently. Pointer network is employed to calculate answer boundaries after representing question-aware passages with match-LSTM by Wang and Jiang [8]. Inspired by Wang and Jiang, an end-to-end NN which is named as gated self-matching network is proposed by Wang and Yang [9] for RC and QA.

Here, we proposed an ALBERT (A Lite BERT) over match-LSTM model to make Xiaoman more intelligent, which is evaluated on NQuAD; the state-of-the-art or comparable performance is accomplished by the proposed model.

Our model is constructed with four modules. First is the preprocessing layer with CWS (Chinese word segmentation) and GloVe [10], which converts the words to vectors. Second is to take the vector as input for the encoding layer utilizing ALBERT. The multi-head attention in ALBERT project the queries, keys and values time linearly, attending to information from different subspaces at different locations. After that, ALBERT is carried out along with "catch important features and distributions" multi-head attentions with accurate match-LSTM decoder model to make our model efficient and precise, as illustrated in Figure 1. The code and the pretrained models are available at https://github.com/stwWang/nercita/tree/master/ALBERT_over_matchLSTM (accessed on 6 October 2020).

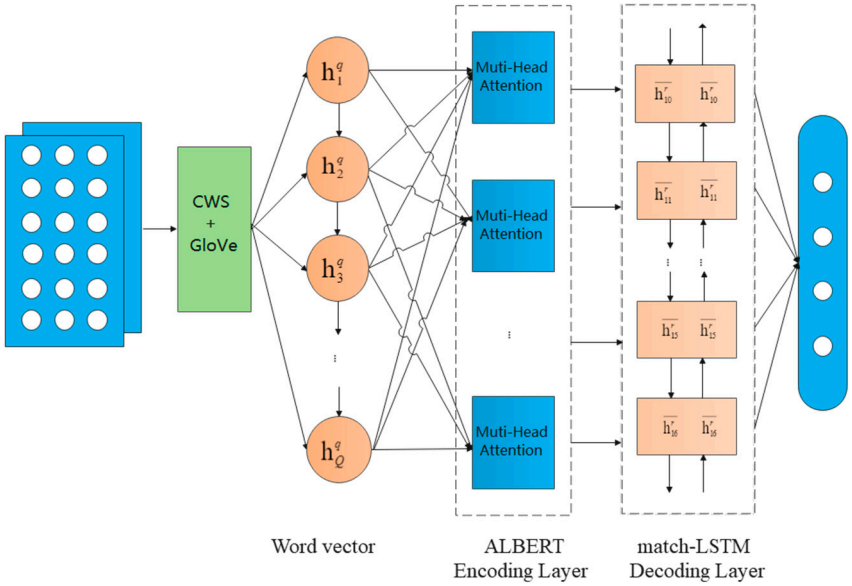

Figure 1. The architecture of our model.

1.2. Related Work

The word embedding technology is derived by the Google open-source project word2vec [11,12] from 2013. Currently, word2vec and GloVe are the most common technologies for vectorization. Both are fundamentally similar: they capture local co-occurrence statistics and distance between embedding vectors. GloVe is count-based; it captures global co-occurrence statistics and requires upfront pass-through entire datasets. A large corpus can be used in many research domains, such as information retrieval [13] and question answering [14–17]. We loaded Sogou agricultural vocabulary as a word segmentation corpus instead of the main word base, which improved agricultural vocabulary recognition greatly.

The Learning to Rank Model uses the machine learning method to solve rank targets. There are three kinds of learning to rank models: the first one is a point-wise model which treats single documents as training objects, the second one is a pair-wise model which uses document pairs to train objects and the last one is a list-wise model, which maintains the whole document list as an optimized object, using the listNet model to optimize the performance.

In recent years, notable results have been achieved by deep learning models in RC and QA. Within natural language processing, there is a stereotype that word vector representations are often involved in much of the work through neural language models, since computer cannot take the words either Chinese [18] or non-Chinese directly as the input for the ALBERT model.

The QA system comprises question classification, search, text retrieval, answer extract, answer ranking and output and text classification technology will be employed to classify the questions into different types based on the answers, which is one task for this paper.

Non-parametric techniques have been studied and used as classification tasks such as K-Nearest Neighbor (KNN) [19]. Support Vector Machine (SVM) [20–22] is another popular technique which employs a discriminative classifier for document categorization. Some well-performed kernel-based models utilizing local features filtered by Convolutional Neural Networks (CNN) layers [23,24] on the practically important sentence classification [25–29] have been revealed.

Many NLP systems take words as atomic units, such as the N-grams model, and the simple systems trained on a massive amount of data outperform the complex model trained on a reduced amount of data. However, the simple model is not sufficient in many tasks. Google has published ALBERT [30] with parameter-reduction techniques, which outperforms BERT. For example, the amount of relevant in-domain data for automatic speech recognition is limited—the performance is usually dominated by the size of high-

quality transcribed speech data (often just millions of words). Thus, to acquire significant progress, we should focus on more advanced methods.

2. Model Architectures

Our end-to-end architecture for text classification is illustrated in Figure 1. The model consists of four parts: text preprocessing layer, ALBERT layer, match-LSTM layer and interactive classification layer. For pre-processing, we employ an efficient and precise model proposed by Duan et al. [31] for Chinese word segmentation (CWS). Based on the stereotype in most NLP tasks, attention is strong in catching the "important" features and distribution of the input, employing multi-head attention. A recurrent network encoder is engaged to build representation for questions and answers separately; thus, we pre-train the data with the ALBERT model, which is a lighter and faster version of BERT, as it cuts down the parameters to reduce the memory consumption and speed up the BERT training model. After that, the match-LSTM model will be used to find the correct category of the question (as illustrated in Figure 1 in more detail), by collecting information from questions and answers.

2.1. Chinese Word Segmentation

As Chinese text cannot be taken directly as the input for the classification model, it is necessary to convert them into vector. To preserve as much as possible the integrity and comprehensiveness of the semantic meaning of a sentence, we first preprocess the sentence with de-noising and word segmentation and vectorization, and then use the GloVe method to transform the word segmentation results into word vectors.

Duan et al. report a Chinese word segmentation (CWS) model—called attention is all you need for CWS (AAY_CWS)—which achieves state-of-the-art performance. Compared with Python's Jieba word base, Duan's CWS model is a more advanced greedy decoding segmentation algorithm, which employs a transformer-like method—Gaussian-masked Directional (GD) transformer. For smoother training, it consumes two highway connections: one is GD multi-head attention and the other is GD attention. The Python implementation of AAY_CWS can be found at https://github.com/akibcmi/SAMS (accessed on 6 October 2020).

The segmentation results of Chinese sentences are greatly influenced by semantics and context. To improve the precision of segmentation, the stop words table is loaded before segmentation, which can reduce the adverse effect of the disabled words, special characters and spaces with little or no contributions to feature extraction in the sentence.

The attention mechanism has achieved great success in many fields. Most competitive neural sequence transduction models have an encoder and decoder layer. Using attention to connect the encoder and decoder performs well, since attention represents the relationship between words. The encoder maps the input sequence denoted as X to a sequence Z; given Z, the decoder then produces an output sequence Y.

An attention model maps a query vector Q and a set of key–value (K, V) vector pairs as (Q, K, V) to an output vector. As shown in Figure 2, we can visualize the attention relationship between words with colors. The deeper the color, the closer the relationship between the word "wishes" and others. As shown in the table, "they" has the deepest color with "wishes", which means the model finds that "they" refers to "wishes".

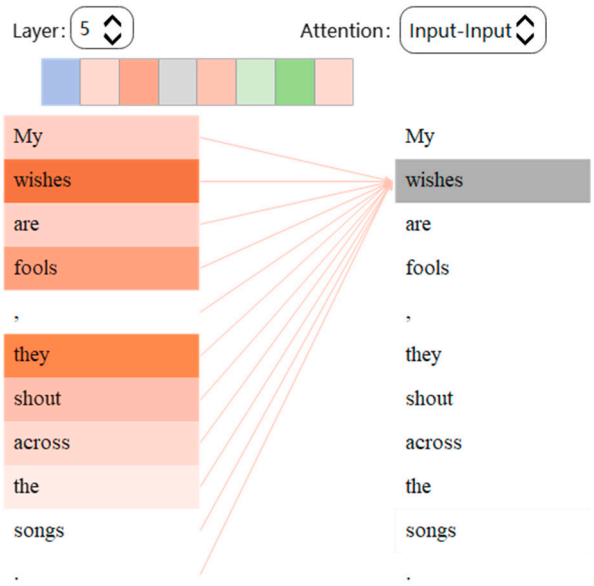

Figure 2. Attention visualization.

2.2. Global Word Vector

Semantic vector space models of language represent each word with a real-valued vector, and these vectors are useful as features in a variety of applications, such as information retrieval [12], document classification [32], question answering and named entity recognition [33,34].

Therefore, based on our wide-coverage agricultural corpus, we employ GloVe for the global log bilinear regression model properties needed for such regularities. GloVe stands for Global Vectors, which uses global words statistics information.

First, GloVe provides annotation for word–word co-occurrence counts with matrix X, whose entries X_{ik} represents the number of times word i occurs in the context of probe word k. We take the final formula directly from the GloVe source:

This is one example of an equation:

$$P_{ik} = P(k|i) = X_{ik}/X_i \tag{1}$$

P_{ik} represents the probability of word k occurring in the context of probe word i.

First, noting that the ratio P_{ik}/P_{jk} depends on three words i, j and k, the most general model takes the form:

$$F(w_i, w_j, \widetilde{w_k}) = \frac{P_{ik}}{P_{jk}} \tag{2}$$

After a few variations, Pennington et al. proposed a new weighted least squares regression model that addresses these problems. Casting Equation (3) as a least squares problem and introducing a weighting function $f(X_{ij})$ into the cost function gives us the model:

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} - log(X_{ik}))^2 \tag{3}$$

The higher the co-occurrence of (j, k), the bigger the weight X_{ik}, the assumption is reasonable—except for high-frequency words. Some high-frequency auxiliary words without actual meanings should be ignored. Therefore:

$$f(x) = \begin{cases} \left(\frac{x}{x_{max}}\right)^{\alpha}, & if\ x < x_{max} \\ 1, & Otherwise \end{cases} \tag{4}$$

where $\alpha = 3/4$ and $x_{max} = 100$.

Take sheep and peach from our dataset as examples. Table 1 shows these probabilities and their ratios for a large corpus, and the numbers confirm these expectations. Compared to the raw probabilities, the ratio is better able to distinguish relevant words (raise and pick) from irrelevant words (disease and radio) and it is also better able to discriminate between the two relevant words. This co-occurrence result shows that our word vectors are good word vectors, preserving the relevant features of words.

Table 1. Two non-discriminative words in our dataset.

Probability and Ratio	k = Raise	k = Pick	k = Disease	k = Radio
P(k/sheep)	0.6×10^{-3}	0.7×10^{-5}	3.1×10^{-3}	1.4×10^{-6}
P(k/peach)	2.3×10^{-5}	6.5×10^{-3}	6.8×10^{-3}	1.3×10^{-6}
P(k/sheep)/P(k/peach)	26	10^{-4}	0.46	1.08

2.3. ALBERT

In 2018, Google proposed the BERT model [35], which set a record in the 11 task tests at that time, bringing a landmark change to the development of natural language processing.

BERT is a seq2seq [36] model of encoder–decode structure, while ALBERT is an improvement based on it. There are four methods to turn BERT to ALBERT. Figure 3 shows the structure of factorized embedding parameterization.

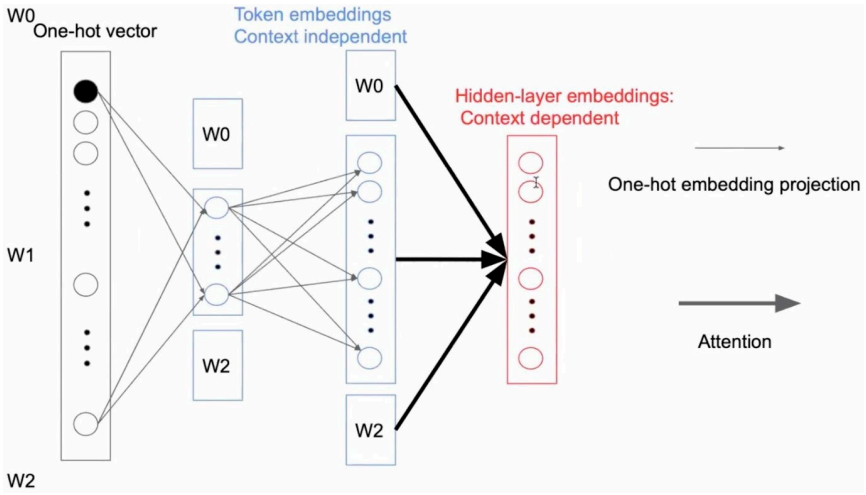

Figure 3. One way BERT turns to ALBERT is by factorized embedding parameterization.

It can be seen from Figure 3 that the encoder of the model contains two layers. The first layer is the multi-attention network layer, which can extract different features of the model, and the second layer is the feedforward network layer. Each layer contains a function of concatenation and standardization of input and output information.

The calculation formula of multi-head attention is as follows:

$$M = Concat\ (h_1, \ldots, h_n)W^o \tag{5}$$

In the above formula, W^o represents the additional weight of Formula M, *Concat* represents the logical concatenation and h_i represents the attention mechanism, as follows:

$$h_i = Attention\left(QW_i^Q,\ KW_i^K, VW_i^V\right) \tag{6}$$

where Q, K, V represent a query vector Q and a set of key–value (K, V) vector pairs as in the Attention model and W_i^Q, W_i^K, W_i^V represent the corresponding weight matrix. The formula of the Attention mechanism is as follows:

$$Attention(Q,\ K,\ V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{7}$$

where k^T denotes the transpose of k vector and d^q is the vector dimension of q. Softmax is a normalization function as follows:

$$SoftMax(z)_i = \frac{e^{z_i}}{\sum_{k=1}^{N} e^{z_k}} \tag{8}$$

As a lighter version of BERT, the ALBERT model transformed Next Sentence Prediction (NSP, ref. [35]) to Sentence Order Prediction (SOP), which improves the downstream effect of multi-sentence input. Compared to BERT, the ALBERT model reduces the number of parameters and enhances the ability of semantic comprehension.

2.4. Decoder Layer

For decoder layer, match-LSTM is employed; to help understand the model, LSTM is explained. The long short-term memory network (LSTM) [37] is a special type of recurrent neural network (RNN), which solves the problems of RNN, such as the disappearance of feedback and the long interval and delay of prediction sequence. The LSTM model includes three gate structures: forget gate, input gate and output gate. The three gates act on the cell unit to form the hidden middle layer of the LSTM model. The so-called gate is actually a mapping representation of the data model, which determines whether to connect or close through the combination operation of sigmoid function and matrix multiplication, and controls whether the current time node information is added to the cell unit. The key to LSTMs is the horizontal line running through the top of the diagram.

The working mode of LSTM is basically the same as that of RNN network, and its network structure is shown in Figure 4. (All the annotation used in Figure 4 is explained in Figure 5.)

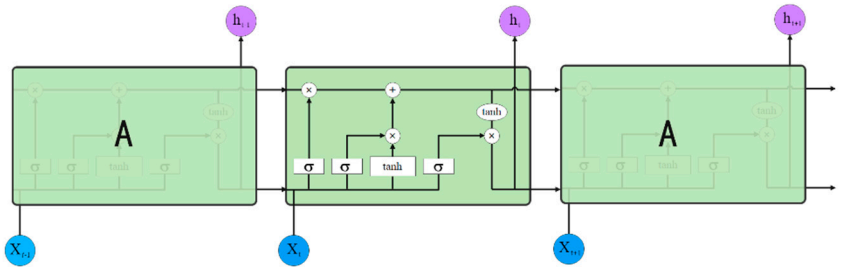

Figure 4. The repeating module in an LSTM contains four interacting layers.

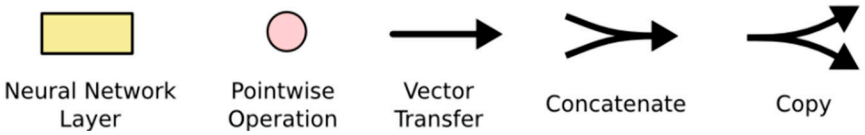

Figure 5. All the annotations in Figure 4.

As can be seen from Figure 4, for the set represented by a listed input sequence x, the calculation formula of hidden state h_t is as follows:

$$h_t = f(h_{t-1},\ x_t) \tag{9}$$

While the "model" in the Figure 4 represents f, which is a non-linear function, h_{t-1} represents the hidden state at time $t-1$. Taking the sample X as input of the model, a

set of hidden state $\{h_1, h_2, \ldots, h_t\}$ will be calculated, where h_t represents the final state of the sequence transferred to the end of the neural network. It can be understood that any input x_t in the middle will interact with the hidden state of the secondary $t-1$, and thus the self-loop form on the left side of Figure 4 can be equivalent to the expression on the right side.

The first step in our LSTM is to decide what information we are going to throw away from the cell state. This decision is made by a sigmoid layer called the "forget gate layer".

$$f_t = \sigma\left(w_f \cdot \left[h_{t-1}, x_t\right] + b_f\right) \tag{10}$$

where w is the weight matrix and b is the bias unit.

The input gate layer is as follows:

$$i_t = \sigma(w_i \cdot [h_{t-1}, x_t] + b_i) \tag{11}$$

The output gate layer is calculated as follows:

$$o_t = \sigma(w_o \cdot [h_{t-1}, x_t] + b_o) \tag{12}$$

The cell state is calculated as follows, where $\widetilde{C}_t$ represents the new memory cell or temporary memory cell and C_t represents the final cell at time t:

$$\widetilde{C}_t = tanh(w_c \cdot [h_{t-1}, x_t] + b_c) \tag{13}$$

$$C_t = f_t * C_{t-1} + i_t * \widetilde{C}_t \tag{14}$$

The hidden state at time t is as follows:

$$h_t = o_t \times tanh(C_t) \tag{15}$$

In 2016, Wang and Jiang proposed a match-LSTM sequence-to-sequence model for predicting textual entailment, which is a widely used and well-explored method for RC tasks. Based on different answer pointer layer, Wang proposed two types of match-LSTM mode: sequence and boundary models. In the case of textual entailment, two sentences, a premise–hypothesis pair, are specified. In our case, a question–answer pair is given. To find whether the answer matches the question, we will practice the boundary match-LSTM model. After the output of ALBERT, the results will be sent to the match-LSTM to get multiple candidates, and the n-best re-rank model will be used. Among them, the top four results (or there could be less or only one result) will be sent to users, and the users will decide which result is the correct one.

The boundary model consists of an LSTM Preprocessing Layer, a match-LSTM layer and an answer Pointer layer; the detailed formula can be found in the paper of Wang and Jiang.

3. Datasets and Experimental Setup

3.1. Data: 20,000 Questions

We use NQuAD to conduct our experiments. Python's regular expression is employed to clean and filter the obtained text data to remove useless information.

We used a random sample of two thousand QA pairs of NQuAD, and these 20,000 peach-related questions were classified into 12 categories (shown in Table 2): marketing, plant diseases and pests, animal disease, cultivation management, breeding management, fertilizer science, nutrition, harvest process, agricultural equipment, storage and transportation, slaughtering process and "OTHERS". Sample questions are shown in Table 3. From Table 2, we can see that the distribution is not a uniform distribution—there are more than 6000 questions for plant diseases and pests, while only 28 for slaughtering process, which is a challenge for our classification task.

Table 2. Distribution of question categories.

	No. 1	**No. 2**	**No. 3**	**No. 4**	**No. 5**	**No. 6**
Category	Marketing	Plant diseases and pests	Animal disease	Cultivation management	Breeding management	Fertilizer science
Count	443	6702	2044	4284	991	840
	No. 7	**No. 8**	**No. 9**	**No. 10**	**No. 11**	**No. 12**
Category	Nutrition	Harvest process	Slaughtering process	Agricultural equipment	Storing and transportation	OTHERS
Count	128	137	28	309	127	3967

Table 3. Sample questions for 12 categories.

ID	Category	Examples
1	Marketing	How much is the market price for peach currently?
2	Plant Diseases and pests	What are the symptoms of peach powdery mildew?
3	Animal Disease	What are the common diseases for mutton sheep breeding? How to prevent?
4	Cultivation management	What is the main strategy for peach tree cultivation in spring?
5	Breeding management	What things should be paid attention to during sheep raising in spring?
6	Nutrition	What are the green foods for ecological chicken farming?
7	Fertilizer science	How to manage the peaches in the greenhouse in terms of water and fertilizer?
8	Harvest process	How to fertilize peach fruit after harvesting?
9	Slaughtering process	What kind of animal diseases are related to the slaughter of pigs?
10	Agricultural Equipment	How to fix tractor flameout in winter?
11	Storing and Transportation	What should be paid attention to when picking, storing and transporting nectarines?
12	OTHERS	What is the medicinal value of okra?

The data are sampled as follows. For each category, 80% of the questions are sampled as a training set (with 16,000 questions), 10% as a test set (2000 questions) and the rest as a validation set (2000 questions).

3.2. Statistics of Our Data

We divide our NQuAD questions into six types, which are common in search logs, as shown in Table 4. The statistics are illustrated in Table 5. We set the dimension of word vector as 120, and max sentence length as 100 words.

Table 4. Six types of example questions.

	Fact	**Opinion**
Entity	When is the peach tree planted?	How to water the peach tree?
Description	How to prune Jingyan peach?	Why should the peach be bagged during growth?
Yes/No	Is 18–23 °C the best temperature for peach leaves growth?	Is long-shoot pruning good for Jingyan peach?

Table 5. Percentage of six types of questions in our NJTG datasets.

	Fact	Opinion	Total
Entity	23.4%	8.5%	31.9%
Description	34.6%	17.8%	52.5%
Yes/No	8.2%	7.5%	15.6%
Total	66.2%	33.8%	100%

3.3. Flow Chart of Data Set Construction

The flow chart of our classification model based on ALBERT+match-LSTM is as follows (Figure 6), which contains input layer, feature extraction and output layer.

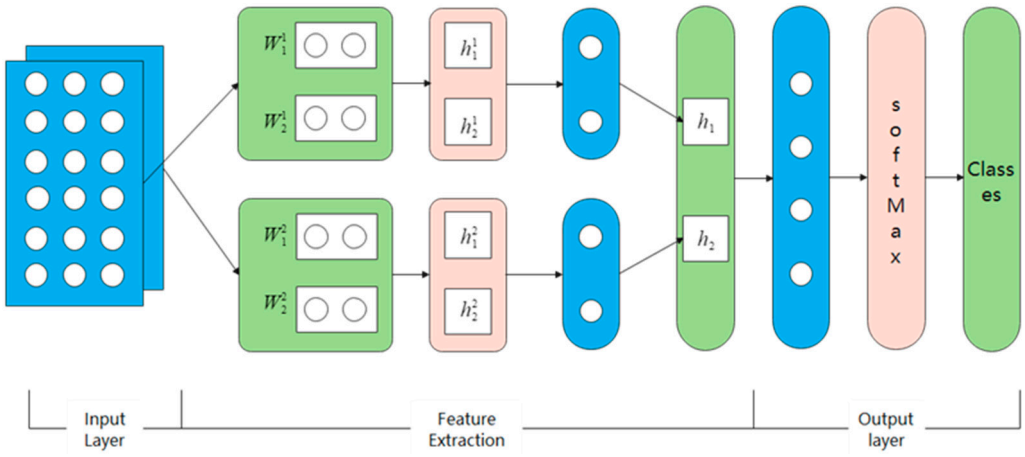

Figure 6. Flow chart of our model.

- Input layer

CWS + GloVe → N*512-dimension word vectors, which is the input for our ALBERT+match-LSTM model.

- Feature extraction layer

In this layer, parameters will be trained to extract the significant features. We set the feature dimension as 512, N = 128, and the initial weight is a Gaussian distribution $X \sim N(0,\ 0.01)$. The K-fold cross validation is employed to train our model and monitor the performance; 16,000 questions were selected as our training set and 2000 as validation set. Set batch size as 1000, in total 16 batches; set the epoch as 500, the sample will be validated every 400 epochs. Adam optimization algorithm is used with $\alpha = 10^{-3}$, $\beta_1 = 0.9$, $\beta_2 = 0.999$. The dropout regularization is employed to avoid overfitting and the gradients are clipped to the maximum norm of 10.0.

- Output layer

After the features are extracted, all the feature vectors will be concatenated to be a one-dimension vector, and finally, the SoftMax function is used to obtain the feature vector output.

3.4. Hardware, Software Environment and Evaluation Indicators

This experiment's software environment is Python 3.6.2 and TensorFlow 1.13.1, the server's hardware environment is NVIDIA Corporation device 1e04 (Rev A1) and GPU is NVIDIA GeForce RTX 2080ti. In this study, the TensorFlow neural network framework is used to construct the neural network.

In the experiment, 20,000 problems are divided into the training set, verification set and test set according to a ratio of 8:1:1. The precision, recall rate and F_1 value are used as evaluation indexes in this paper. The formula is as follows:

$$P = \frac{TP}{TP + FP} \tag{16}$$

$$R = \frac{TP}{TP + TN} \tag{17}$$

$$F_1 = \frac{2PR}{P + R} \times 100\% \tag{18}$$

4. Results

4.1. Chinese Word Segmentation Results

We first tokenize all the questions and answers. The resulting vocabulary contains 100k unique words. We use "attention is all you need" for Chinese word segmentation (CWS)—the result is illustrated in Table 6—and then use word embedding from GloVe to initialize the model.

Table 6. Example of Chinese words segmentation.

Number	CWS	Sentences
1	Before	Peaching bagging can make the peach skin smoother, cleaner, purer and pinker.
	After	Peach, bagging, can, make, the peach skin, smoother, cleaner, purer, and, pinker
2	Before	The best temperature for peach root growth is 18–22 °C.
	After	The best, temperature, for, Peach, root, growth, is, 18–22 °C

4.2. Test Results and Analysis

Attention is used to measure the importance of different labels. From Table 7, the proposed model ALBERT+match-LSTM has a slightly lower recall than Attention+match-LSTM, but it outperformed for precision.

Table 7. Evaluation of different models on NQuAD.

	Type	Models	P (%)	R (%)	F_1 (%)
ML		Multi-SVM	71.3	68.7	70
		KNN	81.7	77.4	79.5
DL	Convolution Kernel-Based	CNN	84.5	81.8	83.1
		LSTM	92.3	93.7	93.0
		match-LSTM	91.7	95.6	95.1
	Attention-Based	Attention-LSTM	92.3	97.9	94.1
		ALBERT+LSTM	92.9	95.4	92.6
		ALBERT+match-LSTM	**96.8**	**97.6**	**96.9**

Compared with other well-known and accepted multi-label classification methodologies, such as multi-SVM, KNN, CNN, LSTM, match-LSTM, attention + LSTM and ALBERT+LSTM, our proposed model has the highest matching performance on NQuAD. Those methods can be classified into classic machine learning (ML) algorithms (SVM, KNN) and deep learning (DL) algorithms, and those DL algorithms can be further divided into convolution kernel-based (CNN, LSTM, match-LSTM) and attention-based (attention + LSTM, ALBERT+LSTM, ALBERT+match-LSTM). Table 7 shows the evaluation of different models on NQuAD.

Table 8 shows that our model has better performance in the plant diseases and pests, animal disease, cultivation management, storage and transportation and "OTHER" categories (highlighted in bold); those five precisions, recall rates and F_1 values of matching question pairs are greater than 96.8%, 97.6% and 96.9%, respectively, and the overall classification effect is better than other models. The F1 value of this model is significantly higher than that of other models in the data sets with fewer data of marketing, nutrition, slaughtering process and four other categories, which indicates that the ALBERT+match-LSTM model can still effectively extract the features of a short text in the case of insufficient data.

Table 8. Evaluation of our model in different categories.

	Categories	P (%)	R (%)	F_1 (%)
1	Marketing	92.2	90.0	91.4
2	Plant diseases and pests	**98.0**	**97.6**	**97.9**
3	Animal disease	**97.2**	**95.7**	**96.8**
4	Cultivation management	**96.8**	**97.8**	**97.0**
5	Breeding management	91.3	94.5	93.3
6	Nutrition	92.7	95.3	94.9
7	Fertilizer science	93.2	92.6	92.8
8	Harvest process	92.6	94.3	93.1
9	Slaughtering process	95.9	93.2	93.4
10	Agricultural equipment	95.7	95.4	95.5
11	Storing and transportation	**97.4**	**98.3**	**97.3**
12	OTHERS	**96.9**	**97.7**	**97.2**

Figure 7 shows the relations between training and validation precision and number of training epochs—a linear co-relation. As the cross validation is used, the training precision is close to validation precision and just slightly higher, which indicates the epoch was good and our model was not overfitting. Before epoch = 4800, these two values are competing. As observed, epoch = 5200 is a turning point. Before the turning point, the precision rate rises steadily with epochs; while after, the precision curve tends to be flat. The training set has the highest precision rate of 97.1% at the turning point, and the verification set has the highest precision rate of 91.9% at epoch = 4800.

Table 9 shows the response time and precision of four neural network models based on attention mechanism on 2000 test sets, which meets the requirements for quick classification of question sentences. CNN is the fastest in response time due to the simple structure of the CNN model, fewer training layers and fewer model parameters. The coherent proposed in this paper, the ALBERT+match-LSTM model, can accurately classify question sentence categories in the test set in 14 seconds and the precision rate reaches 96.8%, which is much higher than that of other models. More interestingly, it is observed that our ALBERT+match-LSTM model is faster than ALBERT+LSTM, with a slight advantage of 2 s in response time; the difference is the decoder layer, match-LSTM vs. LSTM. Our explanation is that the boundary model of match-LSTM performs well in that it only needs to predict two indices, start and end indices, which is more efficient than LSTM.

Table 9. Response time and precision of four network models.

Model	Response Time/s	Precision%
CNN	8	84.5
match-LSTM	11	91.7
ALBERT+LSTM	16	92.9
ALBERT+match-LSTM	14	96.8

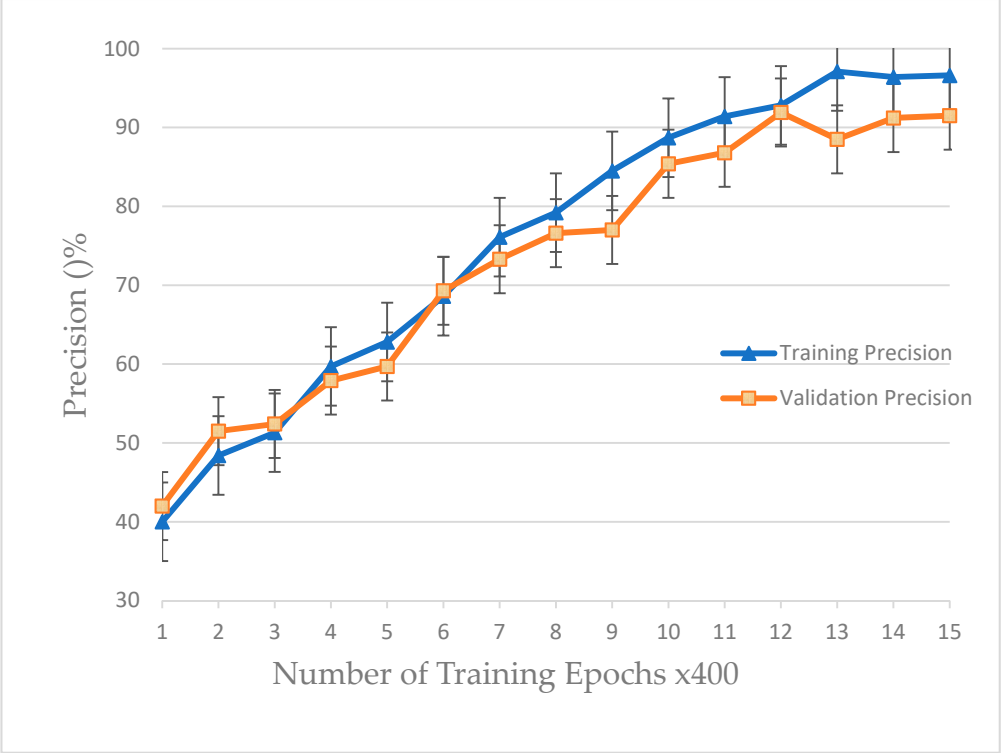

Figure 7. Training and validation precision change with the number of training epochs.

5. Discussion and Conclusions

In this paper, to improve the performance of QA machines, we propose to use Duan's Chinese word segmentation method to preprocess our data, utilize GloVe instead of word2vec to represent the words, and then employ ALBERT as an encoder and match-LSTM as a decoder, which treat the question as a premise and the answer as a hypothesis over multi-head attention (attention score).

Word2vec is predictive, while GloVe is count-based. GloVe can be implemented in parallel, which means it is faster than Word2vec to obtain to a set precision rate. Additionally, GloVe uses global corpus statistics, which contains more information.

In this paper, our attention-based method is compared with classic machine learning models (Multi-SVM and KNN), as well as convolution-kernel models (CNN, LSTM and match-LSTM).

SVM (here, multi-SVM is used) has been one of the most efficient machine learning algorithms since its introduction in the 1990s. However, the SVM algorithms for text classification are limited by the lack of transparency in results caused by a high number of dimensions, while KNN is limited by data storage constraints for large search problems to find nearest neighbors. Additionally, the performance of KNN is dependent on finding function, thus making this technique a very data-dependent algorithm.

CNN, LSTM (a modified RNN) and match-LSTM use multiple convolution kernels to extract text features, which are then inputted to the linear transformation layer followed by a sigmoid function to output the probability distribution over the label space.

ALBERT takes the advantage of attention mechanisms and convolution kernels, thus reaching the highest performance. Extensive experimental results show that the proposed methods outperform other models by a substantial margin (3.9%). Further analysis of experimental results demonstrates that our proposed methods not only find the adequate answers to the question, but also select the most informative words automatically when predicting different answers.

This model can be quite valuable in practice; as mentioned previously in the introduction, this agri-intelligent QA system is built on the NJTG platform. Currently, there are more than 10 million QA pairs collected and stored in the NJTG QA dataset (20,000 were

used in this paper); what is more, end-users of the NJTG platform submit thousands of questions every day. With the application of this model, our QA system can identify the most adequate answers from the exact category (about 100 thousand QA pairs) instead of the whole NJTG QA dataset (10 million QA pairs). In other words, the time complexity decreases 100 times.

Regarding the drawback of this model, if one question sentence "what vitamin can be used for pig's night blindness" can be classified into two categories (nutrition or animal disease), based on our current model, it can belong to only one category; there is a chance that the adequate answer cannot be identified but some constructive work can be done in a sense.

For future work, there is still a gap between the English and non-English (such as Chinese) intelligent QA systems. With the rapid development and high precision of current English QA models and machine translation (MT) models, can we take a detour? To bridge this gap, first there should be a built English-based dataset using MT models from non-English data; there are tons of well-built English-based datasets. Second, we should translate non-English questions into English using high precision MT models. Third, we should identify the adequate answers with a high precision method, and translate the answers back to the original language. It sounds like extra work, but with many ready-to-use models for both tasks, it might be worth the effort. Furthermore, this general method could be applied to all languages, not only Chinese. Based on this grand unification theory, English plays the role of a bridge—all we need are MT models and English-based models.

Author Contributions: Conceptualization, X.W. and H.W. (Huarui Wu); methodology, X.W.; software, G.Z.; validation, Z.L. and G.Z.; formal analysis, X.W.; investigation, H.W. (Haoriqin Wang); resources, X.W.; data curation, G.Z. and Z.L.; writing—original draft preparation, X.W.; writing—review and editing, H.W. (Haoriqin Wang); visualization, Z.L. and G.Z.; supervision, H.W. (Huarui Wu); project administration, X.W.; funding acquisition, H.W. (Huarui Wu). All authors have read and agreed to the published version of the manuscript.

Funding: Supported by Beijing Municipal Committee of Science and Technology (Z191100004019007), supported by China Agriculture Research System of MOF and MARA Grant CARS-23-C06, and National Key Research and Development Program of China (2019YFD1101105). Huarui Wu is the corresponding author of this paper.

Data Availability Statement: The data presented in this study are available on request from the corresponding author. The data are not publicly available due to privacy policy of the authors' institution.

Acknowledgments: This work is sponsored by the Beijing Municipal Committee of Science and Technology (Z191100004019007), China Agriculture Research System of MOF and MARA Grant CARS-23-C06, and National Key Research and Development Program of China (2019YFD1101105). Huarui Wu is the corresponding author of this paper.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Hermann, K.M.; Kocisky, T.; Grefenstette, E.; Espeholt, L.; Kay, W.; Suleyman, M.; Blunsom, P. Teaching machines to read and comprehend. *Adv. Neural Inf. Process. Syst.* **2015**, *28*, 1693–1701.
2. Hill, F.; Bordes, A.; Chopra, S.; Weston, J. The goldilocks principle: Reading children's books with explicit memory representations. *arXiv* **2015**, arXiv:151102301.
3. Rajpurkar, P.; Zhang, J.; Lopyrev, K.; Liang, P. Squad: 100,000+ questions for machine comprehension of text. *arXiv* **2016**, arXiv:160605250.
4. Cui, Y.; Chen, Z.; Wei, S.; Wang, S.; Liu, T.; Hu, G. Attention-over-attention neural networks for reading comprehension. *arXiv* **2016**, arXiv:160704423.
5. Seo, M.; Kembhavi, A.; Farhadi, A.; Hajishirzi, H. Bidirectional attention flow for machine comprehension. *arXiv* **2016**, arXiv:161101603.
6. Xiong, C.; Merity, S.; Socher, R. Dynamic Memory Networks for Visual and Textual Question Answering. In Proceedings of the International Conference on Machine Learning, New York, NY, USA, 20–22 June 2016.
7. Xiong, C.; Zhong, V.; Socher, R. Dynamic coattention networks for question answering. *arXiv* **2016**, arXiv:161101604.
8. Wang, S.; Jiang, J. Machine comprehension using match-lstm and answer pointer. *arXiv* **2016**, arXiv:160807905.

9. Wang, W.; Yang, N.; Wei, F.; Chang, B.; Zhou, M. Gated self-matching networks for reading comprehension and question answering. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, Vancouver, BC, Canada, 30 July–4 August 2017.
10. Pennington, J.; Socher, R.; Manning, C.D. GloVe: Global vectors for word representation. In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), Doha, Qatar, 25–29 October 2014.
11. Mikolov, T.; Chen, K.; Corrado, G.; Dean, J. Efficient estimation of word representations in vector space. *arXiv* **2013**, arXiv:13013781.
12. Ling, W.; Luís, T.; Marujo, L.; Astudillo, R.F.; Amir, S.; Dyer, C.; Black, A.W.; Trancoso, I. Finding function in form: Compositional character models for open vocabulary word representation. *arXiv* **2015**, arXiv:150802096.
13. Schütze, H.; Manning, C.D.; Raghavan, P. *Introduction to Information Retrieval*; Cambridge University Press: Cambridge, UK, 2008.
14. Voorhees, E.M. Overview of the TREC 2001 question answering track. In Proceedings of the Tenth Text REtrieval Conference TREC, Gaithersburg, MD, USA, 13–16 November 2001.
15. Greenwood, M.A. Proceedings of the Coling 2008. Proceedings of the 2nd workshop on Information Retrieval for Question Answering. 2008. Available online: https://aclanthology.org/W08-1800.pdf (accessed on 21 March 2021).
16. Jijkoun, V.; de Rijke, M. Retrieving answers from frequently asked questions pages on the web. In Proceedings of the 14th ACM International Conference on Information and Knowledge Management, Bremen, Germany, 31 October–5 November 2005.
17. Stoyanchev, S.; Song, Y.C.; Lahti, W. Exact phrases in information retrieval for question answering. In Proceedings of the 2nd Workshop on Information Retrieval for Question Answering (Coling 2008), Manchester, UK, 24 August 2008.
18. Cui, Y.; Liu, T.; Chen, Z.; Wang, S.; Hu, G. Consensus attention-based neural networks for Chinese reading comprehension. *arXiv* **2016**, arXiv:160702250.
19. Li, L.; Weinberg, C.R.; Darden, T.A.; Pedersen, L.G. Gene selection for sample classification based on gene expression data: Study of sensitivity to choice of parameters of the GA/KNN method. *Bioinformatics* **2001**, *17*, 1131–1142. [CrossRef]
20. Manevitz, L.M.; Yousef, M. One-class SVMs for document classification. *J. Mach. Learn. Res.* **2001**, *2*, 139–154.
21. Han, E.H.S.; Karypis, G. Centroid-based document classification: Analysis and experimental results. In Proceedings of the European Conference on Principles of Data Mining and Knowledge Discovery, Lyon, France, 13–16 September 2000; Springer: Berlin/Heidelberg, Germany, 2000; pp. 424–431.
22. Karamizadeh, S.; Abdullah, S.M.; Halimi, M.; Shayan, J.; Javad Rajabi, M. Advantage and drawback of support vector machine functionality. In Proceedings of the 2014 International Conference on Computer, Communications, and Control Technology (I4CT), Langkawi, Malaysia, 2–4 September 2014; pp. 63–65.
23. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **1998**, *86*, 2278–2324. [CrossRef]
24. Kim, Y. Convolutional Neural Networks for Sentence Classification. *arXiv* **2014**, arXiv:1408.5882.
25. Kalchbrenner, N.; Grefenstette, E.; Blunsom, P. A convolutional neural network for modelling sentences. *arXiv* **2014**, arXiv:14042188.
26. Huang, W.; Wang, J. Character-level Convolutional Network for Text Classification Applied to Chinese Corpus. *arXiv* **2016**, arXiv:1611.04358.
27. Joulin, A.; Grave, E.; Bojanowski, P.; Mikolov, T. Bag of tricks for efficient text classification. *arXiv* **2016**, arXiv:160701759.
28. Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; Hovy, E. (Eds.) 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, San Diego, CA, USA, 12–17 June 2016.
29. Yang, P.; Sun, X.; Li, W.; Ma, S.; Wu, W.; Wang, H. SGM: Sequence generation model for multi-label classification. *arXiv* **2018**, arXiv:180604822.
30. Lan, Z.; Chen, M.; Goodman, S.; Gimpel, K.; Sharma, P.; Soricut, R. ALBERT: A lite BERT for self-supervised learning of language representations. *arXiv* **2019**, arXiv:190911942.
31. Duan, S.; Zhao, H. Attention Is All You Need for Chinese Word Segmentation. *arXiv* **2019**, arXiv:191014537.
32. Sebastiani, F. Machine learning in automated text categorization. *ACM Comput. Surv. (CSUR)* **2002**, *34*, 1–47. [CrossRef]
33. Turian, J.; Ratinov, L.; Bengio, Y. Word representations: A simple and general method for semi-supervised learning. In Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics, Uppsala, Sweden, 11–16 July 2010.
34. Li, L.; Zhou, R.; Huang, D. Two-phase biomedical named entity recognition using CRFs. *Comput. Biol. Chem.* **2009**, *33*, 334–338. [CrossRef] [PubMed]
35. Devlin, J.; Chang, M.-W.; Lee, K.; Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv* **2018**, arXiv:181004805.
36. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *arXiv* **2014**, arXiv:14093215.
37. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef]