An Identification Method of Question Subjects Based on Word Embedding and LSTM

M X Gao and Z X Fu
Department of Information Science, Beijing University of Technology, Pingleyuan 100, Chaoyang District, Beijing, China
Email: gaomx@bjut.edu.cn

Abstract. Using the subject of the question can locate the question area, narrow the scope of the query, and provide users with better answers. The question text is usually short text. Therefore, in view of its sparse features and irregular structure, this paper proposes an identification method of question subjects based on word embedding and LSTM (IQS-WE-L), and uses question set on the MadSci website for experimentation, which has three subjects. We firstly use the Word2vec to train the Wikipedia database to generate a dictionary. Then based on word vectors, we propose four feature extraction methods: W2V, W2V-TFIDF, W2V-c-TFIDF and W2V-c, which formalizes the text features into vectors through word embedding and other features. Finally, we build an LSTM network for classification training to identify the subject of the question and quantitative evaluate effect of four feature extraction methods we proposed. Experimental data shows that the method proposed in this paper can effectively identify the subject of the question. When classifying the subject of the question, the F1 value can reach a maximum of 0.9339.

1. Introduction
The 45th “China Internet Development Statistics Report” shows that the number of Internet users across the country reached 904 million, and the Internet penetration rate reached 64.5%. The huge number of Internet users constitutes China’s booming market and lays a solid foundation for the development of the digital economy. While the number of netizens is growing rapidly, the increasing amount of information in the cyberspace has brought challenges to information retrieval. How to find the required content in mass information has become one of the focuses of people's attention. Using subject of question can locate the problem area, narrow the scope of the query, and provide people with higher-quality answers. Therefore, it is necessary to study the identification method of question subjects.

The question text is usually short text. The identification of the question subjects can be achieved by short text classification. Academia generally defines short text as text with less than 100 words. It has the characteristics of few words, messy sentence format. Short text classification training can be divided into the following stages: pre-processing of short text data, text representation and feature extraction of short text data, and training and optimization of classification models.

In the stage of text representation and feature extraction, it is more common in traditional methods to use the bag-of-words model for text representation, and then use TF-IDF for feature extraction. Ref. [1] proposed a method, which used the bag-of-words model and TF-IDF to represent text and extract text feature. But this method ignores the interconnection between words. Compared with the long text, the short text has a small number of words and an irregular structure, which makes it difficult to obtain
the key features of the text. If using the traditional bag-of-words model, a high-dimensional sparse vector space will be generated, which will bring difficulties to the subsequent training and reduce the final accuracy rate. At present, researchers mainly solve this problem by introducing external databases to expand short text features. Ref. [2] introduced Wikipedia as an external database to build a feature expansion vocabulary to expand short text features. Ref. [3] proposed a feature extension algorithm based on Wikipedia word vector to improve the accuracy of the text classification.

In the stage of training and optimizing the classification model, the traditional text classification method mainly selects SVM, Naive Bayesian Model, k-Nearest Neighbor algorithm and decision tree for text classification. Ref. [4] classified ship industry news based on SVM. Ref. [5] used decision tree method to classify webpage text containing sensitive words. With the development of artificial neural networks, CNN, RNN, LSTM and GRU have been used in the field of text classification, and achieved excellent classification results. Ref. [6] used the LSTM to classify text. Ref. [7] used Doc2Vec to extract topic information and integrated it into CNN and improved the text classification effect of traditional CNN network. Ref. [8] proposed a model based on RNN to analyse Chinese text sentiment. Ref. [9] proposed a text classification algorithm based on convolutional capsule network.

This paper proposes an identification method of question subjects based on word embedding and LSTM (IQS-WE-L). This method firstly pre-processes the problem set. Secondly, Word2vec model is trained with Wikipedia dataset to construct a dictionary, formalized the words, and expand the semantic features. Then, based on the word embedding, the text features in the question set are extracted. Finally, build an LSTM network for question classification and train it.

2. Identification Method of Question Subjects Based on Word Embedding and LSTM

The process of the method proposed in this paper is shown in figure 1.

![Figure 1. The process of IQS-WE-L.](image)

We firstly introduce the English Wikipedia corpus, and uses the Word2vec to train it and generate a dictionary. Then, we pre-process the collected question set. Secondly, we select feature words and extract text features based on the word embedding generated by Word2vec training. Thirdly, the
problem set is divided into train set, validation set and test set, and corresponding label vectors are generated for it. Then, we design and build a classification model based on LSTM and input the train set into the classification model for training and optimizing parameter. Finally, after the model is trained, test set is inputted into the model, and the prediction results are compared with the labels of the train set to evaluate the quality of the model.

2.1. Word2vec Model Training Based on Wikipedia

Wikipedia was created by Jimmy Wales and Larry Sanger. There are currently 288 language editions of Wikipedia, and more than 30 million articles. Its content covers a wide range of topics including science, technology, humanity, history, geography, society, and culture. At the same time, because Wikipedia articles are manually edited and the text conforms to people's language expression rules, it is often used as an external database in natural language processing projects. Because our question set is in English, we train the Word2vec model with English edition of Wikipedia.

In 1986, Geoffrey proposed the concept of distributed representation and word vector [10]. In 2003, Bengio summarized the deficiencies of traditional statistical language models, designed a framework NNLM (Neural Network Language Model) based on neural networks to establish statistical language models [11], and first proposed the concept of word embedding. The NNLM model has two shortcomings. One is that the model can only handle fixed-length sequences, and the other is that the training speed is too slow. In 2013, the Google team led by Mikolov released the Word2vec model, which improved the NNLM model and solved the above two problems [12]. Word2vec contains two models. One is CBOW (Continuous Bag-of-Words Model) model, the other is Skip-gram model. The CBOW model uses the context to predict the target word and learn the expression of word embedding. The Skip-gram model uses the target word to predict the context. Compared with the CBOW model, the Skip-gram model has a larger calculation scale, and the semantic analysis is relatively more accurate. There are two optimization methods for updating the weight matrix from the hidden layer to the output layer. One is based on the layered Huffman tree, and the other is the negative sampling method.

In the research, we use the open source Word2vec model on GitHub, which has been trained. The model was trained with December 2019 edition of English Wikipedia by Skip-gram and negative sampling optimization. It generated three Word2vec models with different word embedding dimensions. Word embedding dimensions are 100, 300, and 500, respectively.

2.2. Feature Extraction Based on Word2vec

After training Word2vec model with Wikipedia corpus, we filter the feature words, retain the key feature words, and remove the less relevant feature words. We select feature words based on part of speech. Specific content is presented in the experimental part. Based on the Word2vec word embedding, we propose four methods of feature extraction: W2V, W2V-TFIDF, W2V-c-TFIDF and W2V-c.

The first method, W2V, only uses Word2vec word embedding to represent text features. The second method W2V-TFIDF uses the TF-IDF as a weight to modify the Word2vec word embedding. The third method, W2V-c-TFIDF, proposes a category factor $c$ that can represent the relationship between categories, and uses the TF-IDF and $c$ to modify Word2vec word embedding. In the fourth method, only the category factor $c$ is used to modify the Word2vec word embedding.

2.2.1. W2V. This method only uses Word2vec word embedding to represent text features. It combines word embedding of feature words in text and construct a new vector. For a text $d = \{w_1, w_2, ..., w_m\}$, $w_i$ is a feature word of the text $d$, the word embedding of $w_i$ is $v_i = [v_{i1}, v_{i2}, ..., v_{in}]$, and the text feature can be represented as $[u_1, u_2, ..., u_m]$, where $m$ and $n$ represent the number of feature words and the word embedding dimension of the text respectively. The calculation method of $u_i$ is shown in equation (1).

$$u_i = v_i$$ (1)
2.2.2. W2V-TFIDF. This method uses TF-IDF to modify word embedding and represent text features. Although the Word2vec model has expanded semantic features, it cannot distinguish the importance of different words to the text. TF-IDF can represent the importance of a word to a text. TF stands for word frequency, IDF stands for inverse text rate. Therefore, W2V-TFIDF uses TF-IDF to modify word embedding, which can highlight key feature word. For a text set, \( d = \{w_1, w_2, ..., w_m\} \) is one of the texts, \( w_i \) is a feature word of \( d \), and \( m \) is the number of feature words of the text. The word embedding of \( w_i \) is \( v_i = [v_{i1}, v_{i2}, ..., v_{in}] \), and the corresponding TF-IDF is \( t_i \). And the feature vector of the text \( d_j \) is \([u_1, u_2, ..., u_m]\), then the calculation method of \( u_i \) is shown in equation (2).

\[
\mathbf{u}_i = \mathbf{v}_i \times t_i
\]  

(2)

2.2.3. W2V-c-TFIDF. Although TF-IDF can represent the importance of different words to the text, it ignores the relationship between categories. Therefore, this method proposes a category factor \( c \) that can reflect the relationship between categories, and combines it with the Word2vec word embedding and TF-IDF to represent text features. For a text \( d_j = \{w_1, w_2, ..., w_m\} \), \( w_i \) is a feature word, and \( m \) is the number of feature words in the text. There is a category set \( A = \{a_1, ..., a_l\} \), and the text \( d \) belongs to the category \( a_j \), then the calculation method of the category factor \( c \) of \( w_i \) is shown in equations (3)-(5).

\[
c_{i,j} = \frac{p_{i,j}}{p_{i,j}+q_{i,j}}
\]  

(3)

\[
p_{i,j} = \frac{x_{i,j}}{r_{i,j}}
\]  

(4)

\[
q_{i,j} = \frac{y_{i,j}}{s_{i,j}}
\]  

(5)

\( p_{i,j} \) is the frequency of the word \( w_i \) appearing in the text of category \( a_j \), and \( q_{i,j} \) is the frequency of appearing in the text of other categories. \( x_{i,j} \) is the number of times where \( w_i \) appears in the text set belonging to category \( a_j \), \( r_{i,j} \) is the total number of each word appearing in the text set belonging to category \( a_j \). \( y_{i,j} \) is the number of times where \( w_i \) appears in the text set belonging to other category, \( s_{i,j} \) is the total number of each word appearing in the text sets belonging to other categories. When the frequency of the word \( w_i \) appearing in the category \( a_j \) is higher, and the frequency of occurrence in other categories is lower, its discriminating ability is stronger, and the value of the category factor \( c \) is larger.

Supposed that text \( d \) belong to the category \( a_j \), the word embedding of \( w_i \) is \( \mathbf{v}_i = [v_{i1}, v_{i2}, ..., v_{in}] \), and \( n \) is the word embedding dimension, the corresponding TF-IDF is \( t_i \), and the value of category factor \( c \) is \( c_{i,j} \), then text features can be expressed as \([u_1, u_2, ..., u_m]\), and the calculation method of \( u_i \) is shown in equation (6).

\[
\mathbf{u}_i = \mathbf{v}_i \times t_i \times c_{i,j}
\]  

(6)

2.2.4. W2V-c. This method only uses the word embedding generated by the Word2vec and category factor \( c \) to represent text features. Let there be \( d = \{w_1, w_2, ..., w_m\} \), \( w_i \) is a feature word of the text \( d \), the word embedding of \( w_i \) is \( \mathbf{v}_i = [v_{i1}, v_{i2}, ..., v_{in}] \). There is a category set \( A = \{a_1, ..., a_n\} \), and the text \( d \) belongs to category \( a_j \). If the category factor \( c \) of \( w_i \) is \( c_{i,j} \), then text feature can be expressed as \([u_1, u_2, ..., u_m]\), and the calculation method of \( u_i \) is shown in equation (7).

\[
\mathbf{u}_i = \mathbf{v}_i \times c_{i,j}
\]  

(7)

2.3. Construction and Training of Classification Model Based on LSTM Network

LSTM network has improved the RNN and solve its long-term dependence problem. In the traditional RNN, the hidden layer unit often uses only a tanh function or a ReLU function for operation.
However, in the LSTM network, the hidden layer unit adds input gate, output gate, and forget gate. Besides, it uses cell state to add or modify information. The structure of LSTM hidden layer unit is shown in figure 2.

![Figure 2. LSTM hidden layer unit.](image)

### 2.3.1. Construction of Classification Model.
We use Tensorflow2.0 to build a classification model based on LSTM network. The model structure is as follows. The first layer is a LSTM, containing 100 neurons. The second layer is a fully connected layer, the number of neurons is 50, and its activation function ReLU. The third layer is also a fully connected layer, and the number of neurons is also set to 3, because the number of subjects that our questions set has is 3. Its activation function is the SoftMax function. The output of the classification model is a 3-dimensional vector. Each component of the vector corresponds to a subject, and the subject corresponding to the component with the largest value is the prediction result of the model.

### 2.3.2. Training of Classification Model.
After feature extraction, the dimension of the text feature vector is equal to the number of feature words. Before training, it is necessary to process variable-length text feature vectors and unify their dimension. In the training process, there are many parameters that can affect the classification effect of the LSTM model. It mainly includes learning rate, loss function and optimizer.

The learning rate is a parameter when updating the weights. The greater the learning rate is, the faster the weights update. But when it is too large, gradient explosion may happen, and the loss value may continue to oscillate and fail to converge. However, when it is too small, it is likely to occur overfit, and it may trap the model at the minimum point.

The loss function is used to quantify the difference between the predicted value and the actual value of the model. The loss functions used in classification tasks are usually mean squared error or categorical cross entropy.

The optimizer determines the method of calculating weight updates. There are many types of optimizer, such as BGD, Momentum, RMSprop, Adam and so on. The BGD optimizer will calculate the gradient based on the whole samples at each iteration and update it in the direction of gradient descent. The Momentum optimizer introduces the idea of momentum in physics. During the training process, it will speed up the update of the dimension in the direction of gradient descent which has not
changed and slow down the update of the dimension in the direction of gradient descent which has changed, thereby speeding up convergence and reducing shock. The RMSprop optimizer is an adaptive learning rate method that can automatically adjust the learning rate during training, which obtains better learning results. The Adam optimizer is a combination of RMSprop and Momentum, which has the advantages of both.

In the stage of experiment, we train the LSTM classification model with different learning rate, loss function and optimizer to determine parameter. Figure 3 shows the code for training the LSTM model when learning rate is 0.001, optimizer is Adam, and loss function is categorical cross entropy. Specific parameter changes are shown in the paragraph of experiment.

```python
model.compile(loss='categorical_crossentropy', optimizer='Adam', rate=0.001, metrics=['accuracy'])
history = model.fit(x=data_train, y=label_train, epochs=10, validation_data=(data_validate, label_validate))
model.save('500mn_tf.idf.h5')
```

**Figure 3.** Code of training model.

### 3. Experiment

#### 3.1. Collection and Pre-processing of Experimental Data

In order to test IQS-WE-L, we collected English question set on 3 subjects from the MadSci website. The three subjects are “Earth science”, “Medicine” and “Physics”. Then, we use the NLTK toolkit to pre-process the problem set, the main work includes word segmentation, noise removal, and stop word removal.

Table 1 shows the numbers of questions and words during different stage. There are 8001 questions and 371566 words in the whole questions set before pre-processing. Each question has an average of 46 words, which indicates the questions are short texts. After pre-processing, there are 7722 questions and 179493 words in the whole questions set. Each question has an average of 23 words.

|            | Before pre-processing | After pre-processing |
|------------|-----------------------|----------------------|
|            | Numbers of questions  | Numbers of words     | Numbers of questions | Numbers of words     |
| Earth science | 2340                  | 84773                | 2261                 | 40153                |
| Medicine    | 2048                  | 100086               | 1979                 | 48292                |
| Physics     | 3613                  | 186707               | 3482                 | 91048                |
| Total       | 8001                  | 371566               | 7722                 | 179493               |
| Average     | 2667                  | 46                   | 2574                 | 23                   |

#### 3.2. Experimental Design and Evaluation Criteria

In this paper, we design three sets of experiments and select precision, recall and F1 to evaluate the quality of the classification model.

The first experiment is used to test the influence of learning rate, loss function and optimizer on the classification structure, so as to determine each training parameter. Firstly, we test the training results of four different learning rates: 0.001, 0.01, 0.1, and 1. Then, we test the effect of mean square error and cross entropy as loss functions. Finally, we test effects of different optimizer SGD, Adam and RMSprop separately.

The second experiment was used to test the effect of word embedding dimension and feature word selection on classification results. On one hand, the word embedding is set to 100, 300, and 500 dimensions; on the other hand, the text feature words are selected according to part of speech. One group of data chooses noun and verb as feature word, the other group chooses noun as feature word.
Then we use W2V method to extract features from 6 groups of data, and use LSTM text classification model for training.

The third experiment is used to compare the advantages and disadvantages of 4 different feature extraction methods W2V, W2V-TFIDF, W2V-c-TFIDF and W2V-c.

3.3. Experimental Results and Data Analysis

3.3.1. Influence of LSTM Training Parameter. We firstly compare different learning rate, loss function and optimizer. As shown in figure 4, we evaluate the classification result respectively when the learning rate is 0.001, 0.01, 0.1, and 1. When the learning rate is 0.001, the precision, recall and F1 value reach the maximum. The precision is 0.8611, the recall rate is 0.8720, and the F1 value is 0.8665. The classification result is in the second place when the learning rate is 0.1, and in the third place when the learning rate is 1. The result is the worst when learning rate is 0.01. The experimental data does not show a clear relationship between the learning rate and the experimental result. In the following experiments, we set the learning rate to 0.001, because it produced best results.

As shown in figure 5, we evaluate the classification result when using cross entropy and mean square error as the loss function. The result of using cross entropy as loss function is significantly better than the mean square error. Its precision, recall rate and F1 value are higher than the mean square error. According to the experimental results, cross-entropy is used as the loss function in subsequent experiments.

As shown in figure 6, this article tests the classification results respectively when using the SGD, Adam, and RMSprop as optimizer. The classification result of the Adam optimizer is the best, with precision of 0.8611, recall rate of 0.8720, and an F1 value of 0.8665. All three are significantly higher than the other two optimizers. The second best is the RMSprop optimizer, and the worst is the SGD optimizer. According to the experimental results, Adam optimizer is used in the following experiments.
3.3.2. Influence of Word Embedding Dimension and Feature Word Selection Methods. In this experiment, we evaluate the influence of word embedding dimension and the feature word selection methods on classification results. And we use W2V to extract text features. The experimental result is shown in table 2. When the dimension of the word embedding is fixed, the F1 value of selecting nouns and verbs as feature words is obviously greater than that of using only nouns, which indicates that using nouns and verbs can represent text semantics relatively better. When the feature word selection method is fixed, whether using nouns and verbs as feature words or only using nouns, the maximum F1 value is obtained when the word embedding is 500-dimensional, and the F1 value increases as the dimension of the word embedding increases. This shows that in the range of less than 500 dimensions, the higher the word embedding dimension are, the better the classification results are.


Table 2. Influence of word embedding dimension and feature word selection methods.

| Word embedding dimension | 100   | 300   | 500   |
|--------------------------|-------|-------|-------|
| Feature word selection   | Verb & noun | Noun | Verb & noun | Noun | Verb & noun | Noun |
| Precision                | 0.8459 | 0.8249 | 0.8611 | 0.8299 | 0.8733 | 0.8412 |
| Recall                   | 0.8521 | 0.8331 | 0.8720 | 0.8332 | 0.8677 | 0.8447 |
| F1                       | 0.8490 | 0.8289 | 0.8665 | 0.8316 | 0.8705 | 0.8430 |

3.3.3. Influence of Feature Extraction Methods. In this experiment, we evaluate the influence of different feature extraction methods on classification results.

As figure 7 shown, the W2V-c has the largest F1 value, which can reach 0.9339. The W2V-c-TFIDF is next, and its F1 value is 0.8729. The F1 value of the W2V is close to the W2V-c-TFIDF, but slightly below it, which is 0.8705. And W2V-TFID has the lowest F1 value, which is 0.8331.

![Figure 7. Influence of feature extraction methods.](image)

According to the classification results of W2V-c is better than W2V-c-TFIDF, and the results of W2V is better than W2V-TFIDF, we conclude that when using TF-IDF value as a weight to modify vector, the classification results will get worse. Because TF stands for the frequency of words in this text, and IDF is the inverse text rate, which is only related to the frequency of words in other texts. Although TF-IDF can reflect the importance of a word to the text to a certain degree, it only highlights feature words, which is more important compared with all other texts, rather than the feature words with strong discriminating ability. For those feature words that only occur in a large number in a certain category, and appear less frequently in other categories, they have a strong discriminatory ability and are helpful for classification. However, because they occur in large numbers in a certain category, their IDF value will be small, and their characteristics will be weakened. For those rare words that cannot effectively express semantics, the IDF value will be large, and their feature will be strengthened, which will have a negative influence on classification. Therefore, in the final experimental data, when using TF-IDF, the classification results get worse.
According to classification results of W2V-c is better than W2V, and the results of W2V-c-TFIDF is better than W2V-TFIDF, we can conclude that when using category factor c as the weight to modify word embedding, the classification results will be improved. Because when a word occurs more concentrated in a category and less in other categories, its category factor c is larger, and its feature will be strengthened. These words have strong discriminating ability, which is helpful for classification.

4. Conclusion
How to find useful content from massive information is one of the issues which people currently concerned about. The subject question can be used to locate the problem area, narrow the scope of the query, and provide people with higher-value answers. Therefore, we do the research about identification of question subjects. There usually are less than 100 words in a question text. Question texts often have sparse features and irregular structure, and belong to short text. We propose an identification method of question subjects based on word embedding and LSTM (IQS-WE-L). Firstly, we use the Word2vec to train the Wikipedia database to generate a dictionary. Then based on word embedding, we propose four feature extraction methods: W2V, W2V-TFIDF, W2V-c-TFIDF and W2V-c, which formalize the features of the text. Finally, we build an LSTM network for classification training to identify the subject of the question and quantitative evaluate effect of four feature extraction methods we proposed. From the experimental results, the following conclusions can be drawn. First, in the classification task, the cross-entropy loss function is better than the mean square error loss function, and the Adam optimizer is better than RMSprop and SGD. Second, using nouns and verbs as feature words can represent text semantic feature better than only using nouns, which helps categorize the subject of the question. In the range of less than 500 dimensions, the higher the word embedding dimensions are, the better the classification results are. Third, using TF-IDF to modify word embedding in the feature extraction process will have a negative influence on the classification results, because TF-IDF will strengthen the feature of rare words, and weaken some words, which have a stronger ability to classify. Fourth, the using the category factor c to modify word embedding in the feature extraction process will highlight the feature words with strong discriminating ability, thereby improve the classification result.

However, there are still something to improve in this study. On one hand, the feature word selection method in this study is relatively simple, which is only based on part of speech. We can learn and design better feature selection methods in future experiments to improve the classification results. On the other hand, this article only selects English question sets for experiments, and does not conduct experiments on Chinese question sets. There are lots of big differences between Chinese and English. These differences may result in that IQS-WE-L is not suitable for Chinese questions sets. Besides, in the real life, the question may contain both Chinese and English words, and we can do research about it in future.

Acknowledgments
This work is supported by Beijing Natural Science Foundation (4192007), and supported by the National Natural Science Foundation of China (61202076), along with Department of information Science, Beijing University of Technology (2020JSJKY011).

References
[1] Huang C M and Wang S L 2020 Research on short text classification based on bag of words and TF-IDF Software Engineering 23 1-3.
[2] Fan Y J 2012 Research on Chinese short text classification based on Wikipedia New Technol. Lib. Inf. Ser. 28 47-52.
[3] Lei S, Liu X M and Xu W X 2018 Chinese short text classification based on word vector extension Computer Applications and Software 8 269-274.
[4] Zhu F P and Wang X F 2020 Text classification for ship industry news Journal of Electronic Measurement and Instrumentation 34 149.

[5] Li W 2019 Design of Web Sensitive Word filtering system based on decision tree Computer Knowledge and Technology 16 245.

[6] Wu M Q, Wu J M and Xin W B 2020 Optimization of Word2Vec and LSTM multi-category sentiment classification algorithm Computer Systems & Applications 29 130-136.

[7] Yang R, Chen W, He T, Zhang M, Li R L and Yue F 2020 Text classification method based on convolutional neural network using topic information Journal of Modern Information 40 42.

[8] Yan J, Zhao Z H and Zhao R 2019 Research on social media text sentiment analysis based on machine learning China Computer & Communication 20 44.

[9] Kang Y, Li J Y, Yang Q Y, Cui G R and Wang P Y 2019 Text classification using convolutional capsule network based on dual-channel word vectors Computer Engineering 11 177-182.

[10] Hinton G E 1986 Learning distributed representations of concepts Proceedings of the Eighth Annual Conference of the Cognitive Science Society 1 12.

[11] Bengio Y, Ducharme R, Vincent P and Jauvin C 2003 A neural probabilistic language model Journal of Machine Learning Research 3 1137-55.

[12] Mikolov T, Chen K, Corrado G and Dean J 2013 Efficient estimation of word representations in vector space arXiv preprint arXiv: 1301.3781.