Analysis of Big Data Embedded Mapping Model Oriented to Prediction of Subordinate Relationship

Hui Xu*
Nanchang Institute of Technology, Nanchang 330044 China

*Corresponding author: njxh7981@126.com

Abstract. With the rapid development of information technology, research on big data information has gone deep into all walks of life, and massive amounts of data spew out anytime and anywhere. This paper proposes a domain-subordinate relationship acquisition that combines multiple strategies and word representations. From semi-structured text and unstructured text, the candidate entity upper-lower relationship entity pairs are extracted, and then the obtained candidate upper-lower relationship entity pairs are verified by the support vector machine method to obtain high-quality candidate entity pairs. In the training phase, an exploratory method is designed to search for the optimal solution, and a greedy mechanism is introduced to evaluate the effectiveness of the reinforcement learning agent, so that the virtual node mapping scheme can continuously move towards higher system returns. Evolution has finally achieved the optimal virtual network mapping decision, so that large-scale tasks can be deployed to task processing nodes in the appropriate underlying network to achieve efficient task execution in a big data environment. The experimental results show that the Tard method can effectively avoid model overfitting and improve the task recognition processing ability in the actual application process under the premise of meeting large-scale task requests.

Keywords: big data, large-scale task processing, task deployment, and subordinate relationship prediction

1. Introduction
With the rapid development of mobile Internet, Internet of Things, and cloud computing technologies, the prelude to the mobile cloud era has begun [1]. The prospect of data-driven decision-making is gaining more and more recognition, and the concept of "big data" is gradually being familiar and accepted by people. Big data is not a new technology or new product, but a new phenomenon. The definition of big data should conform to its characteristics, namely huge data scale, various types of data, low data value density and fast data processing speed. At present, the amount of data in scientific research, Internet applications, and e-commerce is showing a rapid growth trend [2]. In the past, the realization of decision-making can be based on guessing or a model built in advance, but now people can rely on the data itself to make reasonable decisions. The change from traditional data to big data is not only reflected in the difference in quantity, but also in the improvement of data quality.

The traditional way of data processing covers data mining, data warehouse and online analysis.
processing. For big data, data is no longer the content that needs to be analyzed and processed. Instead, it is necessary to use special ideas and methods to collect, organize and analyze certain laws of data from massive seemingly chaotic data to support planning and analysis in various fields. The clustering method based on text vector mainly uses the clustering method of the text vector space model. If two entities to be disambiguated have the same concept, they are considered to have a certain similarity [3]. Related scholars have proposed semantic disambiguation of named entities based on extracting information from large encyclopedia collections and Web search results, by maximizing the agreement between extracting context information from Wikipedia and document context [4]. During the process of agreement, the system achieved a very high accuracy rate on news stories and Wikipedia articles. The method based on social network needs the support of knowledge base, that is, entity referent can represent the meaning of referent through the association of other entities. By constructing a social network, entities related to entity referents are connected by edges, so that each entity is represented as a node in the network, and nodes are connected by edges to form a huge network. Next, they use some methods such as random walk algorithm to calculate the distance between two entities and set a threshold. If the distance is lower than the threshold, then the two entities may have the same concept. Relevant scholars have constructed the Babel Net social network and combined Wikipedia and Word Net data to increase the coverage of Babel Net [5]. It has done disambiguation experiments on multiple test sets and achieved very good results. Researchers constructed a topic-based entity representation to calculate the topic distance between the context where the entity refers to the context of the candidate entity [6]. The experiments on the set verify the effectiveness of the method.

This paper uses a web crawler to crawl domain entities from travel texts, and then obtains the candidate hypernym lists of entities from the semi-structured Interactive Encyclopedia and Baidu Baike respectively, and then uses pattern matching and semi-supervised from the unstructured plain text corpus. Methods obtain candidate upper-lower relationship entity pairs, use support vector machine machine learning algorithm to verify the obtained upper-lower relationship entity pairs to obtain high-quality candidate entity pairs, and finally train the word vector model to express the candidate entity pairs in the form of vectors. Carrying out offset clustering, you can obtain entity pairs with high semantic similarity. In view of the problem that some task requests in the training set do not have corresponding labels, we use the policy gradient method and the back propagation algorithm for model training, so that the virtual node mapping scheme obtained each time can continue to evolve towards the direction of higher system revenue, and finally achieve large-scale task deployment in a big data environment. Experimental results show that while satisfying large-scale task requests, the Tard method can effectively avoid the occurrence of model over-fitting during task processing, and improve the ability of task identification and processing in actual applications, which improves the overall system to a certain extent.

2. Construction of the Prediction Model of the Upper and Lower Relationship of Domain Entities Represented by Big Data Embedded Mapping Words

2.1. Obtaining Candidate Relationship Instances

In this section, we obtain examples of candidate upper-lower relations, which are obtained from the core words of the entity, semi-structured interactive encyclopedia and Baidu encyclopedia, and unstructured travel text corpus.

The core word of the entity itself is the easiest and most accurate way to obtain the entity hypernym. It can be seen from the entity itself that the core word is generally located at the end of the word. After the corpus preprocessing segmentation, the last word can be obtained as the core word.

Because of the limited patterns of the extracted upper and lower relationship entity pairs and the small number of entity pairs obtained from semi-structured text, the Bootstrapping-based method is used to obtain more candidate relationship instances from unstructured corpus, search the subordinate relation seed set \( S \) in the corpus (Web) to get some sentences \( K \), and then use these sentences \( K \) to
generate model sentences. They match the patterns in the candidate pattern set in the corpus, and obtain the entities in sentence K as the new seed set.

A certain number of candidate upper-lower relations can be obtained through method steps. Through experiments, it can be seen that data based on multiple strategies can be mutually verified and supplemented, and its accuracy rate is greatly improved compared with a single pattern matching. Although the recall rate is greatly improved through the bootstrapping method, it is due to the pattern. Larger generalization results in no significant improvement in accuracy. Next, this article trains a support vector machine (SVM) by constructing the feature vector of the entity's upper and lower relationship, and then verifies the candidate relationship instances. Figure 1 shows a schematic diagram of obtaining the upper-lower relationship of domain entities represented by the integration of multiple strategies and big data embedded mapping words.

Figure 1 Schematic diagram of obtaining the subordinate relationship of domain entities represented by the integration of multiple strategies and big data embedded mapping words.

2.2. Prediction Model Based on SVM Candidate Upper-Lower Relationship

SVM constructs the optimal hyperplane in the feature space, which maximizes the distance between different sample sets closest to the hyperplane, so as to achieve the maximum generalization ability. Suppose the optimal hyperplane of the binary classification problem is:

$$w^*x + b = 0$$

Among them, $x$ is a multidimensional vector, and $w^*x$ represents the inner product of vector $w$ and vector $x$.

After obtaining the candidate upper and lower relationship entity pairs, the context is selected first. The encyclopedia website has very comprehensive and reliable information. Therefore, the candidate upper and lower relationship entity pairs are queried on the encyclopedia web page, and the first 20 pieces of information are selected as experiments corpus, the search scope is limited to encyclopedia sites. The selection of features is very important. It is related to the ability to train a classification model that can distinguish between upper and lower relationships. Therefore, this paper selects a high-discrimination feature set, including character features, part-of-speech features and word features, and these features are converted into numerical feature vectors that SVM can recognize, so as to train an
SVM classifier to verify candidate upper and lower relationship pairs.

Words are the basic units that make up entities. Words are made up of words and words, and their combination methods and methods are all determined by the machine learning model. For example, "Dian" and "Chi" can be combined into "Dianchi", this article first builds a dictionary:

\[ R = \{ r_1, r_2, r_3, \ldots, r_n \} \]  

where \( r_i \) refers to the words that appear in the entity. All the words that appear in the upper-lower relationship instance are placed in the dictionary \( R \), and each word-based feature vector is expressed as:

\[ V(E) = \{ v_1, v_2, \ldots, v_n \} \]  

Among them, the value of \( v_i \) satisfies the following formula:

\[ v_i = \begin{cases} 
0 & r_i \notin E \\
1 & r_i \rightarrow E 
\end{cases} \]  

For part-of-speech features and word features, this article mainly uses the word segmentation tool ICTCLS of the Chinese Academy of Sciences to preprocess the experiment such as word segmentation, and manually perform part-of-speech tagging.

2.3. Context Vector Representation and Recognition

When the domain knowledge base is automatically constructed, due to the semantic difference of the entity pairs, it is often necessary to manually classify and organize them. Therefore, the first problem is that the artificial classification is time-consuming and laborious, and a good semantic understanding is required. The second is that it is affected by humans and has a relatively low accuracy rate. Therefore, the purpose of this section is to automatically organize and identify the semantic upper and lower relations of entities, so as to cluster semantically similar pairs of upper and lower relations.

Assuming that the upper and lower relation vector \( v \) (superior relation) can be obtained through \( v \) (superior word)-v (lower word) approximation, that is to say, for any word, it can be mapped to its superordinate word through a mapping matrix. Then, through the word vector model, any word is expressed as a word vector form \( x \), and its superordinate words are also expressed as a vector form.

We solve the mapping matrix from lower words to upper words by minimizing the mean variance:

\[ \Phi = \frac{1}{N} \arg \Phi_k \min \sum_{(x,y)} \| y - x\Phi \|^2 \]  

where \( N \) is the number of upper and lower word pairs \( (x, y) \) in the training data. The optimization algorithm uses stochastic gradient descent. Because the upper and lower relations are many-to-many relationships, it is necessary to learn a mapping for each upper and lower relation vector cluster matrix:

\[ \Phi^*_k = \frac{1}{N_k} \arg \Phi_{k^*} \min \sum_{(x,y) \in C_k} \| y - x\Phi_{k^*} \|^2 \]  

Among them, \( N_k \) is the number of entity pairs in the \( k \) th cluster of the cluster \( C_k \).

3. Simulation Experiment and Analysis

3.1. Comparison of Deployment Effectiveness of Big Data Embedded Mapping Tasks

In order to evaluate the effectiveness of the Tard method proposed in this paper in large-scale task deployment, we conduct this set of experiments to verify the performance of the Tard method. Based on the task request sets of three different scales, we use the Tard method to conduct multiple experiments and the average results obtained are shown in Figure 2. We can observe that when the task request scale is 2000, the Tard method has a strong performance advantage in terms of accuracy rate, recall rate and harmonic average evaluation index. These results fully illustrate the performance improvement of the Tard method proposed in this paper in large-scale task deployment, and it can
achieve a more reasonable task deployment plan. Compared with Tard, although the Spread Out method can achieve as much as possible to evenly distribute many tasks to each task processing node at the bottom, it does not analyze the overall task deployment based on the remaining resources and task attributes of the underlying network physical nodes. And then it is difficult to find a reasonable task deployment plan. The Non-Spread Out method tends to use some underlying physical nodes as much as possible for task processing, which is very likely to cause the overload of task processing nodes and the overall load imbalance of the system, and reduce resource utilization. Although the proposed Tard method is based on historical data and deep neural network training, it is still very effective in the identification and processing of tasks other than the training set in practical applications. This is mainly because the current task deployment plan can be analyzed according to the task demand information and the underlying available resource information, and the agent with self-learning and self-evolution capabilities in reinforcement learning can be used to evaluate a virtual mapping plan and make an appropriate choice. It maximizes the benefits of the system while the mission is deployed successfully. At the same time, Dropout is used to alleviate the model overfitting problem and enhance the generalization ability of deep network model. Finally, large-scale tasks in the big data environment can be effectively executed.

![Figure 2](image2.png)

**Figure 2** Tard task deployment effectiveness.

In order to further verify the performance of our proposed Tard method in large-scale task deployment and its adaptability, we conduct multiple sets of experiments and introduce accuracy to verify the accuracy of Tard's task deployment to target nodes in the model application phase. Here we still analyze by observing the performance comparison of the three methods. Figure 3 shows the experimental results of the tasks ranging from 1,000 to 4,000. We can observe that the accuracy of the Tard method exceeds 80%. The accuracy of the Spread Out method and the Non-Spread Out method does not exceed 80%. It can be seen that the Tard method can deploy the task to its target node with a greater probability in order to complete the task and execute it efficiently. This is mainly because the deep computing model we proposed can use the features selected from the task request and the underlying node to learn.

![Figure 3](image3.png)

**Figure 3** Comparison of Tard with Spread Out and Non-Spread Out in terms of task deployment effectiveness.

### 3.2. Comparison of System Revenue Performance

In this set of experiments, we compare the performance of Tard, Spread Out, and Non-Spread Out from the long-term average return of the system. As shown in Figure 4, as a whole, the returns of the
Tard method, Spread Out method, and Non-Spread Out method are all in a fluctuating trend. In this process, the system revenue value obtained by the Tard method is always greater than the revenue value of the other two methods. The overall difference between the Spread Out method and the Non-Spread Out method is relatively small.

![Graph showing comparison of Tard, Spread Out, and Non-Spread Out methods](image)

**Figure 4** Comparison of Tard, Spread Out and Non-Spread Out in the long-term average return of the system.

4. Conclusion

This paper proposes a method for extracting and verifying the subordinate relationship of domain entities that integrates multiple strategies and word representations, including obtaining semi-structured candidate relationship instances, obtaining unstructured candidate relationship instances, and using support vector machines to verify relationship instances. The word vector model represents the upper and lower entity pairs in the form of vectors, calculates their offset vectors, and then performs semantic clustering. Aiming at the problem that some data (task request) in the training set does not have a corresponding label, based on the idea of reinforcement learning, we use the policy gradient method and the back propagation algorithm for model training. In the training phase, an exploratory method is designed to search for the optimal solution, and a greedy mechanism is introduced to evaluate the effectiveness of the reinforcement learning agent, so that the virtual node mapping scheme can continuously move towards higher system returns. Evolution has finally achieved the optimal virtual network mapping decision, so that large-scale tasks can be deployed to task processing nodes in the appropriate underlying network to achieve efficient task execution in a big data environment. The Tard method can effectively avoid model overfitting and improve the task recognition processing ability in the actual application process under the premise of meeting the large-scale task request.

References

[1] Engin, Z., van, D. J., Lan, T., et al. Data-Driven Urban Management: Mapping the Landscape. Journal of Urban Management, vol.9, no. 2, pp. 140-150, 2020.
[2] Chhabra, G. S., Singh, V. P., Singh, M. Cyber Forensics Framework for Big Data Analytics in IoT Environment Using Machine Learning. Multimedia Tools and Applications, vol.79, no. 23, pp. 15881-15900, 2020.
[3] Faroukhi, A. Z., El, A. I., Gahi, Y., et al. Big Data Monetization Throughout Big Data Value Chain: A Comprehensive Review. Journal of Big Data, vol.7, no. 1, pp. 1-22, 2020.
[4] Cammarota, G., Ianiro, G., Ahern, A., et al. Gut Microbiome, Big Data and Machine Learning to Promote Precision Medicine for Cancer. Nature Reviews Gastroenterology & Hepatology, vol.17, no. 10, pp. 635-648, 2020.
[5] Thudumu, S., Branch, P., Jin, J., et al. A Comprehensive Survey of Anomaly Detection Techniques for High Dimensional Big Data. Journal of Big Data, vol.7, no. 1, pp. 1-30, 2020.
[6] Shafqat, S., Kishwer, S., Rasool, R. U., et al. Big Data Analytics Enhanced Healthcare Systems: A Review. The Journal of Supercomputing, vol.76, no. 3, pp. 1754-1799, 2020.