Extraction of Space Domain Entity and Relation via Word Vector Representation and Clustering Method

Zhanji Wei1*, Gang Wan1, Lingyong Huang2, Yao Mu1 and Yunxia Yin1

1School of Space Information, Space Engineering University, Beijing, 101400, China
2China Centre for Resources Satellite Data and Application, Beijing 100094, China

*Corresponding author’s e-mail: weizhanji@seu-edu.cn

Abstract. Knowledge graph has shown great value in search engine, natural language Q&A, recommendation system and other application scenarios in recent years. The basic elements of a knowledge graph are entities and relations therein, so how to automatically extract entities and relations from natural language texts becomes a key issue in knowledge graph construction. In this paper, we propose an unsupervised method to extract space domain entities and relations with the goal of building a space knowledge graph. Firstly, a neural network model is used to extract implicit semantic features of domain words represented by dense vectors from original space domain corpus, and then new entities are discovered by clustering in vector space through a small number of labeled data. By concatenating space domain-specific word vectors and general domain word vectors, universal vector representations of entities are obtained, which include general features and domain features as well. On this basis, semantic vectors of relations between entities are calculated, and more new entity relations can be extracted from the corpus by using semantic vectors of relations. Compared with supervised method, the entity and relation extraction method proposed in this paper only needs a small amount of labeled data, thus is quite suitable for the construction of knowledge graph in space domain where labeled data is rather rare and expensive.

1. Introduction

Knowledge graph is a kind of technology that representing human knowledge in a form of structured triplets (head entity, relation, tail entity) interconnected one another so that machines can calculate and analyze knowledge more easily. Google put forward the concept of knowledge graph in 2012 in order to improve intelligence of its search system, and achieved great success. Since then, with deepening of research, knowledge graph has become one of the most thriving artificial intelligence technologies. At present, the scale of knowledge graph in general domain has exceeded one billion, but the development of knowledge graph in specific domain is relatively backward.

Space is a thrilled domain for human beings. To integrate knowledge graph technology into space domain has the prospect to increase intelligence of space systems, which is very helpful in many space mission such as Mars exploration and deep space exploration. In this paper, space knowledge graph is proposed in order to facilitate a more smart space information system. A main step to construct a space knowledge graph is to recognize named entities and extract relations in space domain.

Named entity recognition and relation extraction are procedures to get entities and relations from natural language texts, through which structured knowledge triplets are formed. There are two main types of methods to extract named entities and relations. One is based on supervised method and the other is unsupervised method. Much attention has been paid to the study of supervised method[1, 2]. Guillaume Lample et al presented a bidirectional LSTMs and conditional random fields model for named entity recognition[3]. Jason C used a hybrid LSTM and CNN architecture to recognize named entity[4]. Dong C et al applied a LSTM-CRF neural network that utilizes both character level and radical-level representations to recognize named entity[5]. Makoto M et al study relation extraction using recurrent neural network LSTMs[6].

Content from this work may be used under the terms of the Creative Commons Attribution 3.0 licence. Any further distribution of this work must maintain attribution to the author(s) and the title of the work, journal citation and DOI.

Published under licence by IOP Publishing Ltd

Journal of Physics: Conference Series 1944 (2021) 012022 doi:10.1088/1742-6596/1944/1/012022
Compared to supervised methods, an unsupervised method has the advantage of being more economical and feasible. Thus in this paper, an unsupervised method is adopted. Especially, a vector representation model of space domain vocabulary is designed, through which space domain-specific entities and relations can be extracted through clustering in certain vector space.

2. Methodology
In this section, we expatiate our unsupervised method. We first introduce the vector representation model designed to get features of space domain vocabulary, and then elaborate procedures to recognize and mine domain-specific entities and relations from raw corpus based on pre-trained vector representation model.

2.1. Vector representation model of space domain vocabulary
In order to extract domain entities and relations from original texts of space domain, firstly, space domain words need to be expressed as vectors as the input of subsequent entity and relation extraction model. In this subsection, we design a vector representation model of space domain vocabulary, through which semantic and conceptual features of the vocabulary can be acquired from the collected original corpus in space domain.

One-hot vector coding method is used in traditional vocabulary representation. However, this method has many defects. One is that the coding is extremely sparse while the dimension size is huge. The dimension size of each word vector is equal to the number of words in the vocabulary while there is only one non-zero element. Besides, all the word vectors are irrelevant, that is, word vectors contain no semantic information. Thus, dense vectors with relatively small dimensions are used to encode words in our study. Each dimension of the word vector measures semantic and conceptual features of the corresponding word in a certain aspect. Artificially defined features is difficult to achieve completeness, thus a neural network model is established to obtain the vector representation of words through training, as is show in figure 1.

![Figure 1. Structure of the vector representation model.](image)

In general domain, there are word2vec, glove, Bert and other word vector models. The data used to train these models are from general corpus. However, space domain is a professional field, and its semantic and lexical features have certain domain specificity.

If we use the general domain model only, a large part of space domain-specific information is not taken into account and the result will not be satisfactory. Therefore, the approach we use is to train a space domain vector representation model and fuse it with the general domain model to get a complete vocabulary representation. The training process is as follows:

1) Collect enough accurate and credible raw texts in space domain to form corpus $S_c$.

2) Obtain domain lexicon $S_d$ from corpus $S_c$. The size of $S_d$ is $n$. Data structure of $S_d$ takes the form of key-value pair, the key is each word of $S_d$ and the value is corresponding index of the word. It
has to be noticed that in this step, the case of words must be distinguished, such as access and ACCESS are two words in $S_s$.

3) Determine input and output of the model. Traverse sentences in corpus $S_c$ as input to train the model. A specific sentence $X$ in $S_c$ can be expressed as:

$$X = \{x_1, x_2, \ldots, x_i, \ldots, x_{\text{end}}\}$$  \hspace{1cm} (1)

where $x_i$ is the vector representation of the $i$th word in $X$ and $x_i \in \mathbb{R}^N$. Compared to general model, there are some differences that $x_i$ is a combination of one-hot representation of a word acquired from $S_s$ together with vector representation of overt features of the word such as whether it is capitalized, or whether it is the beginning or the middle or the ending of a named entity, etc. Therefore, $x_i$ is expressed as:

$$x_i = [x_o, x_f]$$  \hspace{1cm} (2)

where $x_o \in \mathbb{R}^n$, $x_f \in \mathbb{R}^m$ and $m$ is the total number of all known features. The expected output of the model, $Y$ corresponding to $X$ is the $2k$ words on the left and right sides of each word in $X$, i.e., the input-output training data pairs are $(x_i, y_i)$, where:

$$y_i \in \{x_{i-k}, \ldots, x_{i-1}, x_{i+1}, \ldots, x_{i+k}\}$$  \hspace{1cm} (3)

4) Design network of the model. A three-layer neural network model is adopted. The number of neurons in the hidden layer is $M$. Matrix $W_h$ with the size of $M \times N$ and matrix $W_o$ with the size of $n \times M$ represent feature weight matrices of input and output of words respectively.

The output of hidden layer is:

$$Y_{h,i} = \tanh(W_h x_i + a)$$  \hspace{1cm} (4)

The output layer is:

$$Y_{o,i} = W_o Y_{h,i} + b$$  \hspace{1cm} (5)

where $Y_{o,i} = [y_1, y_2, \ldots, y_n]$. The probability of each word is obtained through softmax normalization:

$$P(y_i \mid x_i) = \frac{\exp(y_i)}{\sum_{i=1}^n \exp(y_i)}$$  \hspace{1cm} (6)

Cross entropy is used as the optimal loss function of training:

$$Loss = -\sum_{x_i} \log P(y_i \mid x_i) = -\sum_{x_i} \sum_{j=1}^{2k} \log P(x_i,j \mid x_i)$$  \hspace{1cm} (7)

The optimal value of weight matrices $W_h$ and $W_o$ can be obtained by training the model with random gradient descent method. The ith line vector of $W_o$ is word vector corresponding to the ith word in the lexicon $S_s$.

2.2. Unknown entities recognition model

The obtained word vector contains abundant semantic and lexical information of entities, and all the word vectors constitute a characteristic vector space in space domain. The metric distance between entities with similar semantics must be closer than that between entities with unrelated semantics. The more similar two words are in one respect semantically, the closer they are in the vector space. According to these principles, we can obtain a large number of unknown entities based on a small number of known entities.
For entities composed of a single word, take a known entity as the center of a hypersphere with radius $R$, find out all the words within the radius $R$ from the center of the sphere, then it is very likely that these words represent entities of the same type as the known entity. To improve probability, unknown entities can be retrieved through clustering method. In particular, $T$ is defined as a set of entities of a certain category, such as "satellite" or "rocket". If there are $n$ known entities of the same category $T$, i.e., $E_i \in T, i \in \{1,2,\ldots,n\}$. The entities are represented by word vectors $W_o^1, W_o^2, \ldots, W_o^n$:

$$E_i \leftarrow W_o^i$$

(8)

Where symbol ‘$\leftarrow$’ denotes ‘be represented by’. Define $r_T$ is the maximum value of $p$ norm between two entities within $T$:

$$r_T = \max_{i,j=1,2,\ldots,n} \|W_o^i - W_o^j\|_p$$

(9)

The clustering center of all entities within $T$ is:

$$W_o^c = \frac{1}{n} \sum_{i=1}^{n} W_o^i$$

(10)

For an unknown word vector $W_o^x$ that satisfies:

$$\|W_o^x - W_o^c\|_p < \frac{r}{q}$$

(11)

Then the corresponding entity that $W_o^x$ represents also belongs to $T$, in which $q$ is a confidence index to measure credibility of the acquired entity.

2.3. Entity relation extraction procedure

After the entity is identified, the relation between entities can be mined by using the relation extraction model. The usual relation extraction model needs a lot of labeled data, which is very expensive. Using clustering method, more relations of the same category can be extracted with a little labeled data. The detailed procedure is as follows: firstly, the space domain-specific word vector that constitutes the entity is spliced with the general domain word vector, such as output of the Bert model, to get the universal representation of the entity, $W_u^i$:

$$E_i \leftarrow W_u^i = [W_o^i, W_G^i]$$

(12)

The reason why to add the word vector from general domain is that there are more various corpora in the model training process of the general domain, and its word vector may contain general semantic features that the domain-specific word vector does not have. By joining two types of word vector together, they can complement each other and express more complete information of entities.

With the complete vector representation of entity, relations can be extracted naturally. For a certain type of triple relation $R$ (head, tail), assuming $n$ instances are known. For the $i$th instance, the universal vector representations of head and tail entity are $W_u^h_i$ and $W_u^t_i$ respectively. The vector representation of the relation $R$ can be expressed as:

$$W_R = \frac{1}{n} \sum_{i=1}^{n} \frac{q_i (W_u^h_i - W_u^t_i)}{\sum_{i=1}^{n} q_i}$$

(13)

Where $q_i$ is confidence index of the given entities.

With the vector representation of relation $R$, new entity pairs can be found and extracted readily. As is shown in the figure 2, the head and tail entity are the points in the high-dimensional space, and
the dotted circle represents spherical neighborhood of the head entity or tail entity in the high-dimensional space.

![Diagram of spherical neighborhood](image)

**Figure 2.** Vector representation of relation.

For a new head entity $W_U^h$, the tail entity vector can be predicted as:

$$W_U^t = W_U^h + W_R$$

In the formula, it is not required that both sides of the equal sign are strictly equal, but only satisfies that:

$$\|W_U^h + W_R - W_U^t\| < \varepsilon_R$$

where $\varepsilon_R$ is a minimal value set in advance. Similarly, if the tail entity is known, the head entity can be obtained as:

$$W_U^h = W_U^t - W_R$$

3. **Case study**

The main concepts in space domain include spacecraft, rocket, celestial body and so on. For entities belonging to the same type (e.g. satellites), since they have same relation type with other entities as well as same property type, the contextual words around them in human natural language texts are similar naturally. By training the vector representation model to the full proposed in section 2 based on these space domain-specific literatures, certain dimensions of the word vector will reflect this kind of similarities, which leads to clustering of the same type of entities in certain sub space of the word vector space, as is illustrated in figure 3.

![Diagram of clustering](image)

**Figure 3.** Schematic of clustering in certain sub word vector space.

The exact dimensions depicting the feature of different categories of space entity is usually not known in advance, thus the selection of hyper parameters $q$ in equation(11)(13) is very important.
The larger the value of $q$, the higher the credibility will be, but correspondingly, the number of candidate entities retrieved will be reduced. It is difficult to determine whether the category characteristics of an entity are determined by a single dimension or by multiple dimensions, the value of $q$ can be determined by experiment iteratively.

4. Conclusion

In this paper, we propose an unsupervised method to extract entities and relations in space domain for construction of a space knowledge graph. The key contribution of our method is to design a vector representation model to simultaneously encode implicit and overt features of space domain-specific words into a dense vector space. By clustering word vectors in the vector space through calculation of vector norms, entities of different categories can be recognized and retrieved. A universal representation model integrating general domain and space domain-specific features of word vector is also proposed to extract relations through algebraic operation in the vector space. Compared with supervised methods, the proposed method in this paper has an advantage to be easily extended to the construction of various other domain-specific knowledge graphs where labeled data is either expensive or scarce.

References

[1] Xuezhe M and Eduard H 2016 Proc. of the 54th Annual Meeting of the ACL (Berlin: Association for Computational Linguistics) pp 1064-74
[2] Lin Y, Shen S, Liu Z, Huanbo L and Maosong S 2016 Proc. of the 54th Annual Meeting of the ACL (Berlin: Association for Computational Linguistics) pp 2124-33
[3] Guillaume L, Miguel B, Sandeep S, Kazuya K and Chris D 2016 Proc. NAACL-HLT 2016 (California: Association for Computational Linguistics) pp 260-70
[4] Jason C and Eric N 2016 Transactions of the Association for Computational Linguistics vol 4 (California: Association for Computational Linguistics) pp 357-70
[5] Dong C, Zhang J, Zong C, Masanori H, Hui D 2016 Int. Conf. on Computer Processing of Oriental Languages National CCF Conference on Natural Language Processing and Chinese Computing (Beijing: Springer International Publishing) pp 1-12
[6] Makoto M and Mohit B 2016 Proc. of the 54th Annual Meeting of the ACL (Berlin: Association for Computational Linguistics) pp 1105-16