Research and Application of Domain Knowledge Discovery Based on GPT

Yunsong Tan[1,2]
[1]School of Computer Science and Engineering, Wuhan Institute of Technology, China
[2]Hubei Province Key Laboratory of Intelligent Robot, Wuhan, 430205, China

Keywords: Domain Knowledge, Knowledge Discovery, GPT.

Abstract: With the coming of the information age, domain knowledge has developed rapidly and gradually penetrated into many practical applications. An important challenge facing domain knowledge is how to extract valuable in-formation and knowledge from massive data. This paper takes the knowledge in the field of poetry as the research object, analyzes and extracts the knowledge contained in the field of poetry effectively, improves the relevant algorithm, and improves the poetry generation effect of GPT.

1 INTRODUCTION

Domain knowledge plays an important role in the research of many domain problems, and it is no exception in the subject analysis and content analysis of the domain. Domain knowledge is the basic knowledge applied in the field of subject analysis and content analysis. How to build the domain knowledge base and how to make the domain knowledge base play a better role in the subject analysis and content analysis of the domain has become an urgent research problem. Domain knowledge discovery is a technique to extract information from a large amount of domain data in a concise way. The information extracted is hidden, unknown and has potential application value. With the rapid development of artificial intelligence technology represented by GPT technology and the arrival of the era of big data, it is necessary to constantly learn and innovate in order to remain competitive and adaptable.

People are faced with more and more information, and machine learning algorithms, as a powerful tool, provide an effective means to deal with domain knowledge mining and knowledge management tasks. By using machine learning algorithms, we can find potential patterns and rules and help understand the knowledge behind the data. This paper will introduce how to use machine learning algorithms for domain knowledge mining and knowledge management to provide a foundation for GPT to generate high quality knowledge (Sebastiani F,2002).

Before domain knowledge discovery, it is necessary to preprocess the original text data. This includes steps such as removing noise (such as HTML tags or non-alphabetic characters), converting text to lowercase form, and word segmentation (splitting sentences into words). In addition, in some application scenarios, it is also necessary to perform operations such as stemming and stop word filtering. Text creation is regular, then, through the means of data mining, we can find some insight. Followed the train of thought, this paper will use some domain knowledge discovery methods of poetry text corpus. The original data about poetry corpus are from https://github.com/Werneror/Poetry. Indepth mining and analysis of the poetry corpus basic statistical data is as follows: There are nearly 850,000 poems and 29,377 poetry authors in the poetry corpus. Among them, the fields include "title", "Dynasty", "author" and "Content (poetry)". After data cleaning, 504,443 poems were obtained, accounting for 59.1% of the original database(Li F L , 2020).

Feature extraction is the process of converting raw domain knowledge data into numerical vectors that can be processed by machine learning algorithms. Common feature representation methods include Bag of Words model, TF-IDF (Term frequency-inverse Document Frequency) and Word2Vec. These methods can help capture the key information in the domain knowledge, thus providing the basis for subsequent tasks such as classification and clustering. Based on the above data, a poetry corpus containing popular subject labels is constructed for the

Tan, Y.
Research and Application of Domain Knowledge Discovery Based on GPT.
DOI: 10.5220/0012875600004536

In *Proceedings of the 1st International Conference on Data Mining, E-Learning, and Information Systems (DMEIS 2024)*, pages 5-9
ISBN: 978-989-758-715-3

subsequent poetry subject classification and poetry generation tasks. Based on the domain knowledge discovery and semantic analysis of the above poetry corpus, we hope to get interesting findings.

2 DOMAIN KNOWLEDGE DISCOVERY ARCHITECTURE

Domain knowledge discovery refers to extracting structured or semi-structured information from large-scale domain data through computer technology and natural language processing, and then analyzing, reasoning and predicting it. The main goal is to discover the patterns, relationships, or characteristics that lie behind the domain. Based on machine learning algorithms, domain data classification, clustering, sentiment analysis and other tasks can be realized(Huang, 2013). As for the creation process of domain knowledge, it is like solving an optimization problem. Under certain constraints, domain knowledge should follow certain rules. Creators use words to express their inner sense of reality and strive to achieve the realm of beauty. Knowledge Discovery Architecture is shown in Figure 1. It consists of five modules, cotained Domain Knowledge, Construction Feature, Constrction Corpus , Statistic Analysis and Text Generation Based on GPT .

The implementation path of this paper involves two major components of natural language processing, namely natural language understanding (word segmentation, semantic modeling, semantic similarity, clustering and classification, etc.) and natural language generation (poetry generation and poetry translation).

Given a poem text, how to judge whether this fragment is a meaningful vocabulary by randomly selecting a fragment? If the collocations of the left and right parts of this fragment are varied and rich, and the collocation of components within the fragment is fixed, then this fragment can be considered as a vocabulary. In the specific implementation of the algorithm, the index to measure the richness of the external left and right collocation of fragments is called "freedom", which can be measured by (left and right) information entropy; The fixed degree of collocation within fragments is called "solidification degree", which can be measured by the mutual information of sub-sequences.

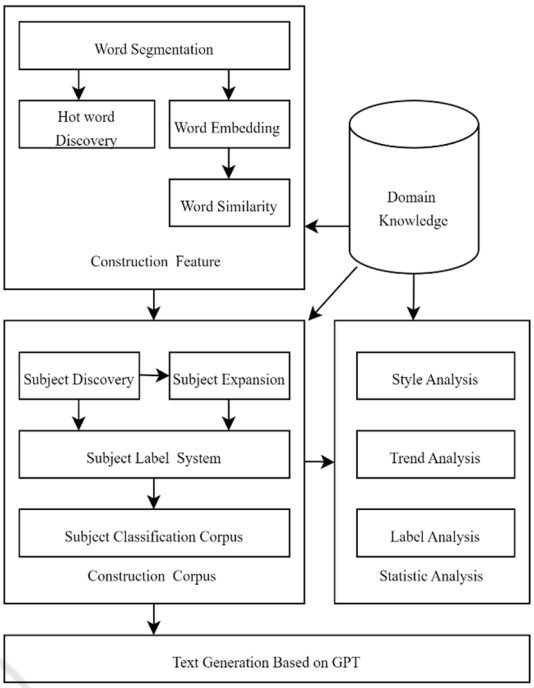

Figure 1: Knowledge Discovery Architecture.

2.1 Train the Word Embedding Model of Semantic Relevance

Word embedding model can automatically learn the correlation between words from massive texts, and thus realize the tasks of word correlation analysis, word similarity analysis, cluster analysis and so on.Word segmentation is the starting point of all subsequent analysis tasks. It is connected with the above high-frequency word mining, combined with the accumulated word bank, and then based on directed acyclic word graph, sentence maximum probability path and dynamic programming algorithm, the word segmentation operation of the 540,000 poems is carried out. After the word segmentation is done properly, it can be input into the word embedding model (in this case, Word2vec) for trainin. Word embedding model based on Word2vec can "learn" from a large number of unlabeled text data to word/word vectors, and these word/word vectors contain semantic correlation between words (can be semantic correlation or syntactic correlation), just as in the real world, "birds of a kind cluster together." Words can be defined by the words around them (the context), and Word2vec word embedding model can learn precisely this relationship between words and context. In the process of training Word2vec, the

model will learn from a large number of poetry text data two kinds of association relationships between words, namely aggregation relationship and combination relationship.

Aggregation relation: Terms A and B have an aggregation relation if they are interchangeable with each other. In other words, if words A and B contain aggregate relations, one can be used to replace the other in the same semantic or syntactic category without affecting the understanding of the whole sentence.

Combinatorial relation: Words A and B have combinatorial relations if they can combine syntactically with each other.

Suppose for a given m poetry documents $\{d_1, d_2, \dots \dots d_n\}$, The number of words contained in document $i(i \in [1, m])$is N_i. After text segmentation and other preprocessing, Word2vec model is used to train the word vector of each word W_{ij}. W_{ij} represents the word vector for the j-th word of the document in Part i.

Set the number of topics to K, Word distribution $\varphi_k = (\varphi_{k1}, \varphi_{k2}, \dots \dots \varphi_{kj})$for each knowledge topic T_k and the subject word vector w_v^k (where $v \in [1, v]$, V represents the number of subject words under each subject), and find the cosine similarity, the calculation formula is as follows:

$$s_{jv} = \frac{\sum_1^n (W_{ij})_t * (W_v^k)_t}{\sqrt{\sum_1^n (W_{ij})_t^2} * \sqrt{\sum_1^n (W_v^k)_t^2}} \quad (1)$$

Where, s_{jv} represents the cosine similarity between the word W_{ij}^k in the text and the word $(W_{ij})_t$ under topic k. $(W_{ij})_t$ and $(W_v^k)_t$ are the components of W_{ij} and W_v^k, respectively, and n represents the dimension of the vector.

After obtaining the similarity between W_{ij} and each subject word W_v^k, the similarity between W_{ij} and topic T_k is:

$$s_j^k = \sum_{v=1}^V \varphi_{kj} * s_{jv} \quad (2)$$

Compare the similarity between a certain word W_{ij} and different topics. If the similarity between W_{ij} and topic T_1 and topic T_2 is close, the word W_{ij} can be used as the relative word of T_1 and T_2 .

2.2 Measure the Semantic Correlation Between Words

Generally, the cosine value between the word vectors of two words is used to represent the similarity or correlation degree between words. The cosine value range of the included Angle is [-1,1]. The smaller the included Angle between the word vectors, the larger the cosine value, and the closer it is to 1, the higher the semantic correlation degree. On the contrary, the correlation is lower.

Based on the above word embedding model, suppose similarity ("soldier", "war") = 0.75, similarity ("soldier", "beacon-fire") = 0.37, similarity ("war", "beacon-fire") = 0.48. Among three words, the semantic correlation between "soldier" and "war" is the highest, followed by "war" and "beacon-fire", and the lowest "soldier" and "beacon-fire". The advantage of this method, which gives a numerical value to identify whether words are related, lies in concise expression and efficient calculation, such as the discovery/clustering of poetry themes to be carried out. However, this calculation of word relevancy does not reflect the "causal path" of word relevancy. So, is there an intuitive way to show the semantic correlations between words and see why they exist in the way they do (that is, to find lexical association paths or semantic evolution paths)? This task of finding the path of lexical semantic evolution needs to be transformed into a TSP problem (traveling salesman problem).

2.3 Use A* Algorithm to Find the Path of Semantic Evolution Between Words

Traveling Salesman Problem (TSP) is one of the most famous problems in the field of mathematics. Suppose there is a traveling businessman to visit n cities, must choose the path to take, the limit of the path is that each city can only visit once, and finally return to the original city. The goal of path selection is to require the path distance to be the minimum value of all paths.

A-star (A* search algorithm) is the most effective direct search method to solve the shortest path in the static road network, and it is also an effective algorithm to solve many search problems. The closer the distance estimate in the algorithm is to the actual value, the faster the final search speed.

Returning to the problem of measuring text relevance, if the shortest semantic evolution route between two words can be found in the word embedding space trained above, the causes and consequences of semantic correlation between these two words can be intuitively presented. The embedding space of word2vec words constructed before is a network structure in which nodes are distributed words and edges are composed of cosine

correlation between words，as shown in Figure 2. The mesh is the word2vec word embedding space constructed above, the nodes Wi(i=1,2,…n) denote the words distributed in it, and the edges cij(i,j=1,2,…n) are formed by the cosine correlation between the word nodes.

It can be seen that the shortest semantic evolution path between two words with weaker semantic correlation (greater distance value) is longer, and the opposite is shorter. Therefore, semantic distance is positively correlated with semantic evolution path length, while semantic correlation degree is negatively correlated with semantic evolution path length. With the previous word embedding model and semantic relevance to "pave the way", the subsequent discovery of popular poetry themes will come naturally.

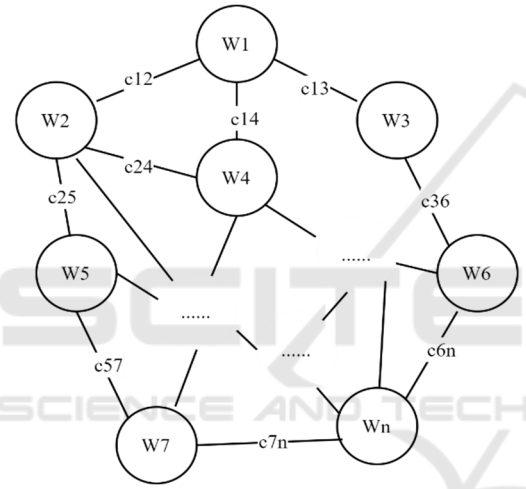

Figure 2: A* Search Algorithm to Find the Path.

2.4 Popular Poetry Themes Based on Community Discovery

In the social network, each user is equivalent to each point, and the whole online interpersonal network is formed by mutual concern between users. In such a network, some users are more closely connected, and some users are more sparsely connected. Among them, the more closely connected part can be regarded as a community, and the nodes within it have relatively close connections, while the connections between the two communities are relatively sparse. The community based discovery algorithm is to mine the large "circles" at the head of the lexical semantic network. If words are personified, the similarity/correlation between words can be regarded as the degree of intimacy between words. Then, the task of discovering poetry subject matter can be regarded as finding a "circle" composed of different members, and the characteristics of the circle can be determined according to the characteristics of the members, in other words, the name of the subject matter can be drawn up according to the connotation of the aggregated words.

3 KNOWLEDGE DISCOVERY BASED ON GPT

3.1 Generate Text Knowledge Based on GPT

GPT (Generative Pre-Training) is an unsupervised language model capable of generating coherent paragraphs of text that has achieved leading performance in many language modeling task benchmarks, such as data scale and parameter scale. Moreover, the model can achieve preliminary reading comprehension, machine translation, question answering and automatic summarization without task-specific training. The core idea can be summed up as the more parameters and samples given, perhaps the better the model can understand natural language and begin to learn to solve different types of NLP tasks without any supervision." The "generation" of GPT generation model is not "water without a source" or "a tree without a root", it is to acquire certain "creative techniques" after fully learning and absorbing some data of predecessors, so that it can generate texts with reasonable effects. At the same time, we can also find some rules of text creation from the generated results, and do some indepth research. In the text generation task, a GPT model of text generation is trained from zero to one to learn various explicit and recessive features in the text data set.

3.2 Comparison of Human-Machine Text Creation Differences

The general principle of text generation modeling is that through a large number of poetry corpus, the poetry generation model can learn the dependency between adjacent words in any poem. For example, when a certain word appears, GPT will guess which word will appear next according to the learned experience, and these words will be stored in the memory of the GPT model in the form of probability. In general, the machine creates text by choosing the word that has the highest probability of appearing in the past, and so on, until it hits the terminator, and gradually generates the entire text. This is the

simplest situation, the resulting effect is very general, and many times it is unreasonable. In order to ensure the generation effect, some complex generation strategies are generally used, such as Beam Search, Top-k sampling, Top-p sampling (Nuclear sampling), Repetition_penalty, Length_penalty, etc., in this way, some other factors of text generation, such as fluency, richness, consistency, etc. will be taken into account, and the effect of text generation can be greatly improved.

4 CONCLUSIONS

Domain knowledge discovery generally has the connotation of "discovery, search, induction and extraction", and the contents to be sought are often not obvious, but hidden in the text, or people can not directly find and summarize in a large range. If you want to pull out the pieces, you need to combine domain knowledge (such as the common sense of poetry in the paper), use a variety of analytical methods (such as various NLU and NLG methods in the paper), and sometimes even need to reverse thinking (such as the generation of poetry in the paper), and it is best that all kinds of analysis should be a sequential and complementary organic whole. In this way, the exploration of text data can be completed with the highest efficiency. Machine learning algorithms provide powerful tools for text mining and knowledge management. By choosing the algorithm model and feature representation method reasonably, valuable knowledge and information can be obtained quickly and accurately. However, it is also necessary to pay attention to the training data quality, feature selection and model evaluation. With the continuous development of technology, machine learning algorithms will play an increasingly important role in the field of text mining and knowledge management, bringing us more applications and discoveries.

ACKNOWLEDGEMENTS

Supported by Science and Technology Research Project of Education Committee of Hubei Province (Grant NO.B2022062).

REFERENCES

Sebastiani, F., 2002. Machine learning in automated text categorization. *Journal of ACM Computing Surveys*.

Li, F, L. , Chen, H. , Xu, G., et al, 2020. AliMe KG: Domain Knowledge Graph Construction and Application in Ecommerce. CIKM '20: *The 29th ACM International Conference on Information and Knowledge Management. ACM*.

Huang, S., Wan, X., AKMiner., 2013. Domain specific knowledge graph mining from academic literatures. *International Conference on Web Information Systems Engineering*. Springer, Berlin, Heidelberg.

Kim, H. Howland, P., Park, H., 2005. Dimension reduction in text classification with support vector machines. *Journal of Machine Learning Research*.

Yang, Y.,1999. An evaluation of statistical approaches to text categorization.*Journal of Infor-mation Retrieval*.

Apte, C., Damerau, F., Weiss, S., 1998. Text mining with decision rules and decision trees. In *Proceedings of the Conference on Automated Learning and Discovery, Pittsburgh*,USA.

Robertson, S, E., Harding, P., 1984. Probabilistic automatic indexing by learning from human indexers. *Journal of Documentation*.

Sarah, A., Alkhodair, Benjamin,C.M., Fung,Osmud Rahman, Patrick, C.K., Hung, 2018. Improving interpretations of topic modeling in microblogs. *Journal of the Association for Information Science and Technology*.

Qiao, B., Fang, K., Chen, Y., et al, 2017. Building thesaurus based knowledge graph based on schema layer. *Cluster Computing*.

Joachims, T., 1997. A probabilistic analysis of the Rocchio algorithm with TFIDF for text categorization. *International Conference on Machine Learning*.