Upgrade and Optimization of Natural language Technology of unmanned Distribution car by using Mathematical Model

Zhen Gong, Danhong Chen*, Yu Li, Qiuning Song and Meilin Li
School of College of Economy and Management, Shenyang Aerospace University, Shenyang, Liaoning Province, China

*Corresponding author e-mail: chendanhong@stu.sau.edu.cn

Abstract. The term "artificial intelligence (AI)" consists of two words, "artificial" and "intelligence". The word "artificial" means artificial, while the word "intelligence" means the ability to think (for example, learning, reasoning and problem solving). Therefore, it can be said that artificial intelligence is an artificial thinking ability. Among them, mathematics plays a decisive role in the research and development of artificial intelligence [1]. At present, the main areas of artificial intelligence development are deep learning, natural language processing, computer vision, intelligent robot and so on. Here, the ability of artificial intelligence to learn, reason and solve problems, the ability to perceive the environment and take action, and the ability of machine learning to improve algorithms automatically are all inseparable from mathematics. This paper will use mathematics to optimize the natural language technology of the unmanned distribution intelligent car.

Keywords: Optimization Of (AI), Mathematical the Model of Artificial Intelligence, Computer Natural Language Processing

1. Introduction

The development of AI artificial intelligence is in full swing in the past two years, various places have set off a craze of artificial intelligence investment, and a large amount of capital has also been injected into this field, in the "China artificial Intelligence Development report 2018": the output of AI papers in China ranks first in the world [2]. In terms of patents, China has become the world's top artificial intelligence patent holder, slightly ahead of the United States and Japan; In the industrial, China has the world's second largest AI company. Beijing has the highest concentration of AI enterprises in the world. In terms of venture capital, China's investment and financing in the field of artificial intelligence accounts for 60% of the world's investment and financing, making it the most "money-absorbing" country in the world. These are also enough to see the great potential of artificial intelligence AI [3]. In this paper, a number of mathematical models will be used to optimize the field of natural language technology of unmanned distribution intelligent car in order to improve the performance of the car in an all-round way.

2. Mathematical Model and Natural Language
The development of natural language processing can be divided into two stages: from the 1950s to the 1970s, scientists' understanding was limited to the way people learn language, that is, using computers to simulate the human brain. The results almost entered the second stage after the 1970s, that is, based on mathematical models and statistical methods[4]. A substantial breakthrough has been made. In the 1950s, the academic understanding of artificial intelligence and natural language understanding was as follows: in order for machines to complete speech recognition, computers must understand natural languages. Because that's what humans do. This methodology is called "bird flying school", which is to see how birds fly to build airplanes. In fact, people invented airplanes by aerodynamics, not by bionics. So how can we understand natural language?

General requirements: parsing statements, that is, through syntax. These grammatical rules can be easily described by computer. Get the semantics. Semantics are more difficult to express in computers than grammars[5].

After 1970, statistical linguistics brought natural language processing back to life, and the key task in it was Giarini and his IBM Watson Lab. At the beginning, they used statistical methods to increase the speech recognition rate from 70% to 90%, while the scale of speech recognition increased from hundreds of words to tens of thousands of words.

Therefore, in order to enable the machine to process natural speech, the key is to establish a mathematical model for the context-sensitive characteristics of natural speech, which is the statistical language model (StatisticalLanguageModel), which is widely used in machine translation, speech recognition, print recognition, spelling error correction, Chinese character input and literature query.

To use the language model, This article needs to know all the conditional probabilities in the model, which This article calls the parameters of the model. Through the statistics of the corpus, the process of obtaining these parameters is called model training. As we said before, we only need to count the number of simultaneous occurrences of two adjacent characters and the number of separate occurrences of $\{w\}$, and then calculate the ratio. However, there is one case that we do not consider if two nearby words do not appear at the same time, namely how to handle $\text{left} \{\{washi1\}, \{wimpi\}\}\text{right}=0$,

whether that means the probability is 0. Of course not, it involves the reliability of statistics. In mathematical statistics, This article dares to use data to predict probability because of the theorem of large numbers, which requires sufficient observations. That is, if sample is too small, it is of course unreliable to use the number of times to predict the probability [6].

So how to train a language model correctly? The direct way is to increase the amount of data. However, This article will still encounter the problem of zero probability, which is called "unsmooth". For the probability that is not smooth, we can not think that the probability of its occurrence is zero, and we can assign a small proportion to these unseen events from the total probability. As a result, the total probability of the events you see will be less than 1, so you need to reduce the probability of all the events you see [7]. As for how much smaller, it should be carried out according to the method of "the more unreliable statistics, the more discounts for it".

Let's talk about it in detail by counting the probability of each word in the dictionary. Suppose there are $\text{Number}$ words in the corpus, and $\text{SN}$ denotes the size of the corpus. $n = \sum \limits_{r=1}^{\infty} \text{number}$ $\text{This means that the number of Sr$ words appearing in each word is multiplied by the number of times they appear. When Sr$ is relatively small, indicating that there are not enough occurrences, a smaller number of times should be used when calculating their probabilities. For example, $\text{dwindrackers} = \text{lefts} (r+1) \text{right}) \frac{\text{Naturr}}{\text{Sm}} \sum \limits_{r} \text{destir}$ $\text{Naturr} = \text{generally speaking. The number of words that appear once is more than that of twice, and the number of words that appear twice is more than that of three times.}$

That is, the larger the number of occurrences Sr$, the smaller the number of words $\text{Number}$, so $\text{dumb0} > 0$ $\text{[8].}$
In this way, for words whose frequency exceeds a certain threshold, their probability estimate is the relative frequency in the data corpus, and for words whose frequency is below the threshold, the probability estimate is also less than their relative frequency[9].

For the binary model

\[
P(W_{i1}|W_{i-1}) = \begin{cases} 
F \left( W_{i1}, W_{i-1} \right) & \text{if } (W_{i-1}, W_{i1}) \geq T \\
F_g \left( W_{i1}, W_{i-1} \right) & \text{if } 0 < (W_{i-1}, W_{i1}) < T \\
Q \left( W_{i1} \right) \cdot F \left( W_{i1} \right) & \text{otherwise}
\end{cases}
\]

Where $T$ is a threshold, usually around 8:10. $f_{gt}(\cdot)$ represents the relative frequency after smoothing, while $Q(w_{imur1})$ ensures that all frequencies add up to 1. This method was proposed by Katz. Another method called Katzbackff method is the deletion of difference method, which uses linear interpolation of low order model and high order model for smoothing, but because the effect is worse than Katz's, it is rarely used.

3. Application of unmanned Distribution Car

3.1 Optimization of Car Natural Language, Using Hierarchical Log-Bilinear Mathematical Language Model

Based on the Log-bilinear language model, Mnih et al proposed a hierarchical HLB (Hierarchicallogbilinear) language model to replace the matrix multiplication with the highest computational cost in the three-layer neural network architecture proposed in reference. On the basis of ensuring the effect, the speed is improved [10].

This idea of stratification was first proposed by Morin et al. They use the IS-A relationship in the WordNet to convert them into a binary tree and then perform classification forecasts. The experimental results show that although the speed has been improved, the performance has declined, and the loss seems to outweigh the gain. Mnih et al drew lessons from the idea of hierarchy, but in the experiment, a bootstrap learning (Bootstrapping) method was used to automatically construct a balanced binary tree and use it to replace the last layer of the network. In predictive vector classification, non-leaf nodes in the binary tree are used, and the leaf nodes constructed at the end of the model are used to identify specific words.

Complexity is also reduced from $O(|V|)$ to $O(\log_2 \text{training})\ (|V|)$. At the same time, a large number of training parameters are involved in the model proposed in reference. A recurrent neural network language model (RNNLM) was proposed by Mikolov et al. to reduce the number of training parameters. It adopts BPTT(Time Back Propagation) optimization algorithm and achieves better results than the optimal method. In subsequent studies, Mikoto and others made various improvements to the RNNLM, including speed and accuracy.

The feedforward network training principle used in the previous method is basically the same as the feedforward network of the antecedently method, but the framework of the two is very different. The structure of the former is roughly shown below.
(a) Abstract representation of neural network architecture

(b) Transfer process of neural network

Figure 1. Two or more references

● (a) is the nonobjective representation framework of the network. Because cyclic neural networks are mostly used in time series, the input layer, veiled layer and export layer all have timing parameters $t$. The hidden layer valuation formula is expressed as follows:

$$s(a) = \text{sigmoid}(Uw(a) + Ws(a-1))$$

● (b) represents the flow process of a cyclic neural network. Whenever a new word is input, the cyclic neural network jointly inputs the word vector of the new word with the state of the previous hidden layer, calculates the next hidden layer state, and repeatedly calculates all the hidden layer states. Finally, the output final result of each veiled layer is procured by the feedforward network.

Unlike taking $n$ words to approximately predict the window mode of the next word, recurring neural networks can really get enough of that information to predict the next word. In fact, this method has both advantages and disadvantages. Optimization must be done in practice to avoid loss of long-distance information. In order to reduce the complex computation from the last hidden layer to the output layer, Mikolov's team adopted a new grouping method: the feature method based on word frequency.

Divide the $|V|$ word into a root $(|V|)$ group, first determine the group to which the next word belongs through the judgment of "$V$" | "$V$" $(|V|)$, and then find out the elements that belong to the group through several judgments. Finally, the average complexity is about $O(|V|)$, than Mnih and Hinton model proposed by the complexity of $O(\log(|V|))$. And the biggest asset of this method is that the configuration is relatively simple, can effectively reduce the error driving device.

3.2 Optimization of Car Natural Language by Using Improved Mathematical Model Based On Word Vector

In 2008, Collobert and Weston first proposed a special word vector computing method. They systematically summarized much natural language processing tasks based on word vectors, such as part of speech tagging, named entity identification, phrase identification, semantic role signation and so on. Different from the $n$-ary grammar model which finds the approximate solution, their word vector training method directly solves the approximate solution. Give the definition 5.

**Definition 5.** The definition $f(wt-n+1, wt-1, wt)$ denotes the score of consecutive $n$ words in the window. $F$ only has a relative high or low score, which does not represent the characteristic of probability. The higher the $f$ score, the more normal the sentence, and the lower $f$ score indicates that the sentence is unreasonable. In extreme cases, if several words are randomly stacked together, the $f$ value will be expressed as a negative score.

Based on this, Collobert and Weston use the Pair-wise method to train word vectors. Among them, the objective function needs to be minimized as follows.
max \{0,1 \cdot f(x)+f(x(w))\}

In the formula, X is a continuous N-p-phrase in the training set. D is the entire dictionary. X is an active sample, X(W) indicates a negative sample, the function f(x) is the score transformation of the active sample, f(x(w)) is the score transformation of negative samples. In the first sum enumeration calculation in the above formula, all the n-ary phrases in the training corpus are selected as positive samples, and all the negative samples are constructed through the second enumeration of the dictionary to get. X(w) means to replace the middle word of the normal phrase x with w, and the final result is that the phrase is definitely not correct in most cases, which can be used as a negative sample. As can be seen from the above formula, the final score of the positive sample is at least 1 point higher than that of the negative sample.

The structure of f function is basically consistent with the network structure in the literature. They are common to: (1) The words corresponding to the window N words are connected to form a long vector; (2) The hidden layer is only calculated by the layer network. The difference is that only one node in the output layer of the Collobert and Weston models represent the score, while the reference model has |V| nodes; in addition, Hard Tanh is used instead of the tanh activation function to reduce the computational complexity.

In the Collobert and Weston models, the window n value is set to 11, and the dictionary size is set to |V| 1300000. After training with Wikipedia English corpus and Reuters corpus for 7 weeks, we get the ClearW word vector. Compared with other word vectors, the main characteristics of CumberW word vectors are as follows:

- (1) the clocked W word vector contains only lowercase words. That is, unlike other word vectors that treat uppercase and lowercase words separately, the thesaurus is not case-sensitive and treats words as lowercase words.
- (2) Use semi-supervised learning to obtain the effect of clearing the word vector. For example, a speech tag portion and a naming entity identification method, which is different from the other method than the non-supervised learning. The word vector trained by word2vec toolkit has a good analogical (Wordanalogy) feature and can represent the semantic and grammatical properties of words to some extent. The knowledge graph-oriented representation learning algorithm TransE is vitalized by this category of ratio. The knowledge graph contains a large number of entities, semantic categories of entities and relationships between entities, which can be represented by triples. The TransE algorithm regards the relationship in the triple as the translation from subject to object, so that the triple satisfies the linear transformation. By using feature representation vector to describe entities and relations, the semantic relations between entities can be calculated more easily.

Sort a set of words and divide the words into two luxate subsets of equal size according to the sort order. Even if it leads to an imbalance in the tree. It is achieved by assigning words to components that are more responsible for words. The third rule, which is also the most complex rule, is an extension of the second rule, which is modified to assign a point to the two components when the two responsibilities are both less than 0.5 within. This rule is designed to generate multiple codes for words that are difficult to cluster. This article will call the algorithms that invests these rules BALANCED, ADAPTIVE and ADAPTIVE (rules), respectively. Finally, as a benchmark for comparison with the above algorithms, this article will use an algorithm to generate random balanced trees. It starts with the random arrangement of words, and then constructs the left subtree based on the first half of the word recursively, and the right subtree based on the second half of the word. This article calls this algorithm RANDOM. The average code length is the sum of the code length associated with the word, which is the average of the word distribution in the training data. The runtime complexity of the hierarchical model is linear in the average code length of the tree used. The average code number of words is the average code of each word in training data distribution. Because each non-leaf node in the tree has its own feature vector, the number of free parameters associated with the tree is linear. The experimental results are as follows:
Table 1. Three Scheme comparing

| Tree Label | Generating algorithm | Average code length | Average number of passwords | Number of non-leaf nodes |
|------------|----------------------|---------------------|-----------------------------|--------------------------|
| T1         | RANDOM               | 14.2                | 1.6                         | 17963                    |
| T2         | BALANCED             | 14.3                | 1.6                         | 17963                    |
| T3         | ADAPTIVE             | 16.1                | 1.0                         | 17963                    |
| T4         | ADAPTIVE(0.25)       | 24.2                | 1.3                         | 22995                    |
| T5         | ADAPTIVE(0.4)        | 29.0                | 1.7                         | 30296                    |
| T6         | ADAPTIVE(0.4)x2      | 69.1                | 3.4                         | 61014                    |
| T7         | ADAPTIVE(0.4)x4      | 143.2               | 6.8                         | 121980                   |

4. Conclusion
Finally, a simple summary is made. Compared with the traditional word vector, using the above mathematical language model for data pre-training, the level of a sentence can be expressed in context, so that the large-scale monolingual corpus can be fully utilized. And polysemy can be modelled. To the greatest extent, it alleviates the dependence of specific tasks on the model structure and improves the accuracy of language recognition.

Acknowledgments
This research was supported by the college students' innovative entrepreneurial training plan of Shenyang Aerospace University under grant S202010143007. Danhong Chen is the corresponding author and instructor of this paper.

References
[1] Erhan D, Bengio Y, Couville A, Manzagol P A, Vincent P, Samy B. Why does unsupervised pre-training help deep learning?. Journal of Machine Learning Research, 2010, pp.625–660.
[2] Sun Zhi-Jun, Xue Lei, Xu Yang-Ming, Wang Zheng. Overview of deep learning. Application Research of Computers, 2012, 29(8): pp.2806 - 2810.
[3] Bengio Y. Learning deep architectures for AI. Foundations and Trends® in Machine Learning, 2009, 2(1), pp.1–127.
[4] Hinton G E, Osindero S, Teh Y W. A fast learning algorithm for deep belief nets. Neural Computation, 2006, 18(7),pp.1527–1554.
[5] Hinton G E, Salakhutdinov R R. Reducing the dimensionality of data with neural networks. Science, 2006, 313(5786),pp.504–507.
[6] Bengio Y, Lamblin P, Popovici D, Larochelle H. Greedy layer-wise training of deep networks. In:Proceedings of the 2007 Advances in Neural Information Processing Systems 19(NIPS’06). Vancouver, Canada:MIT Press, 2007,pp.153-160
[7] Ranzato M A, Poultney C, Chopra S, LeCun Y. Efficient learning of sparse representations with an energy-based model. In:Proceedings of the 2007 Advances in Neural Information Processing Systems 19(NIPS’06). Vancouver, Canada:MIT Press, 2007,pp.1137-1144
[8] Weston J, Ratle F, Collobert R. Deep learning via semi-supervised embedding. In:Proceedings of the 25th International Conference on Machine Learning (ICML’08). New York, USA:ACM Press, 2008,pp.1168-1175
[9] Srivastava N, Mansimov E, Salakhutdinov R. Unsupervised learning of video representations using LSTMs. In:Proceedings of the 32nd International Conference on Machine Learning (ICML’15). Lille, France:Omni Press, 2015,pp.843-852
[10] Jia K, Sun L, Gao S H, Song Z, Shi B E. Laplacian auto-encoders:an explicit learning of nonlinear data manifold. Neurocomputing, 2015, pp.250–260.