Text Modeling Based on the Analysis of Categorical Words

Xiaoli Wang
No.113 XueFu Road, KunMing City, YunNan Province, China
Email: 2931678760@qq.com

Abstract. Recently the most essential research in machine learning is to discover correct topics from large scale of electronic documents, and many studies about text modeling have been made in natural language process with the environment of big data. This paper discusses a fundamental way for people to form words while talking and describing ideas, the document is considered as words convergence and a new method for text modeling based on the relational analysis about verbs, nouns, and modifiers is proposed. Additionally the paper introduces a new algorithm for estimating parameters in the generative probabilities of the model, which can alleviate the process about statistics-based text analysis from computational complexity. Experimental results show that the method improves the quality for learning a document.

1. Introduction
With the popularity of Internet and computer technology, large amount of electronic documents have been accumulated and the underlying information from these texts has become a treasure of inestimable value. If data from these documents can be organized and transformed into an efficient text model, the machine learning to categorize, research, and extract topical keywords is available. So it’s great potential for people to explore important information or to get abstracts from the large scale of corpus.

Considerable efforts have been made in text modeling, Vector Space Model(VSM) [1] was proposed by Salton in 1969, its basic idea is to transform the texts into vectors of identifiers, then the information from texts can be computed by values in the vectors. The definition of values is usually the TF-IDF [2] weighting, so it neglects the semantic message from texts. On the other hand, VSM usually get a high dimensionality by the number of distinct words occurring in a document, and the values in the vector space are always sparse.

Topic model is a statistical model for discovering hidden “topics” in documents[3], and convergences of particular words in documents would be considered to represent a certain topic. Nowadays, topic model is widely used for abstracts from journals or newspapers, and also used to analyze changes in topics over time.

There are two ways for topic modeling, Probabilistic Latent Semantic Analysis (PLSA) [4] considers a document as “bag of words”, then for each document, a latent class which is viewed as representations of “topic” is chosen conditionally to the document, and the word is generated from a single topic, thereby the document can be reduced to a probability distribution on different topics. Formally, the latent topics in the model can be learned by EM algorithm.

Latent Dirichlet Allocation (LDA) [5] is proposed to discover topics in large scale of corpus, each document can be a mixture of various topics, and each topic is characterized by a distribution over words. The words is exchangeable in the document, then the text model can be represented as a three-
level hierarchical Bayesian model, in which each class is modeled as a finite mixture over a set of topics, and each topic is modeled as an infinite mixture over a set of topic probabilities.

Recently, the studies on text modeling are mostly focused on how to recognize the latent topics in documents efficiently with less cost of computational complexity and storage space even in the big data phenomenon. This paper presents a new method for text modeling based on the lexical semantics. In section 2, the relationship among verbs, nouns, modifiers is discussed and a probability of sentence can be generated. In section 3, we describe in detail the process of text model. In section 4, we report the experimental results by the measurement of Perplexity on a public dataset. Conclusions and ideas for future work are given in section 5.

2. Basic Concepts

2.1. Semantics and Information
This paper considers a document as language convergence for people to express some topics. In general, there are always many topics in one document, in order to express ideas more clearly, we are accustomed to describing them by sentences, and every sentence can express part of idea which is relevant to the topics.

From the perspective of Computational Linguistics[6], a sentence can be viewed as a particular logical expression. Then every word in a sentence is a computational symbol, and a word is regarded as the basic unit in a document. When people attempt to write down a topic, they have to organize the words, so words about the topic will appear in a document more or less frequently[7]. Studying the convergence of these particular words may capture some abstract notions, therefore, these words are semantic code for the topic.

Since the essence of language is information coding, it can express the existence characteristics about the objects, and also can express the motion characteristics about the objects. However, a single word can only represent either existence or motion. If we need to describe a topic completely, there are at least two words with different nature in a sentence[8].

\[ \text{Noun} + \text{Verb} \] or \[ \text{Noun/ Adjective(Adverb)} + \text{Verb} \] \hspace{1cm} (1)

With the change of the times, the development of language has become more and more mature, and the sentence structure has been more diverse than shown in formula (1), types and quantities of vocabulary have also been greatly expanded. However, the fundamental logical mode of people thinking about the world is still in the form of “Ontology-Attribute”, consequently the language we use is basically in accordance with “Reference-Attribute”, that is, specifying the describing object, then describing the characteristics of the object. Therefore, we can use a notional word as the center word, and then state its attribute characteristics with other words, in this way, a message can be expressed. If a sentence is regarded as a complete unit of description or information expression, all the words with actual meaning in a sentence is a vocabulary-centered word aggregation, the order of the words is not important.

2.2. Generative Probability of Sentence
There are many words in a sentence, we can divide these words into three categories: verbs used to describe dynamic changes, nouns used to describe things regarding the changes, and adjectives used to modify the nouns or the verbs.

Then the process of forming a sentence is considered as: firstly, we write out the dynamic changes of things, then we mark the objects involved in the actions, lastly we modify the attributes of the actions or objects.

Assuming that there is a document \( D \) that can be segmented, and parts of speech of the words in the document can be specified to form three sets, the set of nouns denoted by \( n = \{n_1, n_2, \ldots, n_k\} \), the set of verbs denoted by \( v = \{v_1, v_2, \ldots, v_r\} \), the set of adjectives and adverbs denoted by \( adj = \{adj_1, adj_2, \ldots, adj_s\} \).
Then every sentence $S$ in the document is a triple made up of the different kinds of vocabulary in the document:

$$S = \{ v_{1-t} \in D, n_{1-s} \in D, \text{adj}_{1-t} \in D \}$$

On the basis of formula (1), we can deduce the generative probability of statement $S$ as:

$$P(S) = P_v(n, v, \text{adj})$$  \hspace{1cm} (2)

And according to the chain rule\[9\] of conditional probability, the formula can be:

$$P_v(n, v, \text{adj}) = P_{\text{adj}}(v, n)P_v(n|v)P_v(v)$$  \hspace{1cm} (3)

Formula (3) is important not only for it simplifies the generative probability with three parameters, but also for it illustrates the interpretation of text model: The probability of generating a sentence is divided into three steps: the verb is chosen as a center word from the vocabulary set, then the nouns which involved in the dynamic change, also known as the objects of the action, are selected, finally, the modifiers are chosen conditionally on the dynamic change or the objects.

3. Text Modeling

3.1. Probability Resolution

In terms of formula (3), the semantic model established in this paper is referred to as a hierarchical probability model with three-levels. When estimating the model, we can assume that the vocabulary within a “word bag model” where words are relatively independent for each other. This property will facilitate the parameter estimation in the following work. Therefore, when we describe the reference of “verb-noun”, we can simplify the formula (3) as follows:

$$P_v(n|v) = \prod_{v \in S, n \in S} P(n|v)$$  \hspace{1cm} (4)

As for the semantic relationship between modifiers and verbs/nouns, when the words are relatively independent, we can attain:

$$P_{\text{adj}}(v, n) = P_{\text{adj}}(v)P_{\text{adj}}(n) = \prod_{v \in S, \text{adj} \in S} P(\text{adj}|v) \prod_{n \in \text{adj}, \text{adj} \in S} P(\text{adj}|n)$$  \hspace{1cm} (5)

Taking the products of the probabilities of words, we obtain the generative probability of a sentence:

$$P_v(n, v, \text{adj}) = \prod_{v \in S, \text{adj} \in S} P(\text{adj}|v)P(\text{adj}|n)P(n|v)P(v)$$  \hspace{1cm} (6)

As a result, the generation probability of sentence can be realized by the conditional probability of just two words in vocabulary set.

There are many sentences in a document, and a complete meaning for a human to express is just based on the grouping of verbs, nouns and modifiers in a sentence, then we can get the generative probability of a document $D$ with triple words:

$$P_D(n, v, \text{adj}) = \sum_{S \in D} P_v(n, v, \text{adj})$$  \hspace{1cm} (7)

Due to the chain rule, the $P_D(n, v, \text{adj})$ can also be decomposed, then based on formula (6) and (7), we can obtain the following calculations:

The probability of a verb is just as selecting one item from all:

$$P_D(v) = \frac{1}{\text{count(verbs)}}$$  \hspace{1cm} (8)

Where $\text{count(verbs)}$ means the number of verbs in the document.

Given a verb, the probability of a noun relevant to the verb in a sentence can be:

$$P_D(n|v) = \sum_{S \in D} \frac{P_v(n, v)}{P_v(v)} = \frac{\text{count}(v \in s \text{ and } n \in s)}{\text{count}(v \in s)/\text{count}(s)}$$

Then we get:

$$P_D(n|v) = \frac{\text{count}(v \in S \text{ and } n \in S)}{\text{count}(v \in S)}$$  \hspace{1cm} (9)
In the same way, the calculation formulas can be attained:

\[
P_D(\text{adj}|v) = \frac{\text{count(v as and adj as)}}{\text{count(v as)}}
\]

\[
P_D(\text{adj}|n) = \frac{\text{count(n as and adj as)}}{\text{count(n as)}}
\]

### 3.2. Parameter Estimation

In order to calculate the conditional probabilities in the above calculations, it is necessary to count the numbers of words appearing in a sentence at the same time. A common method is to build a matrix, since the probability involves three kinds of words, the word probabilities are parameterized by a three-dimensional matrix of \( n \times v \times \text{adj} \), and the values in the matrix is represented by the times of words appearing in the same sentence.

However, it’s complicated to estimate parameters from such a three-dimensional matrix because the large amount of vocabulary in a document may lead to computationally intractable problems. Thus we consider an algorithm which can reduce this three-dimensional matrix by decomposition with one kind of word so that the computation can be available from a normal two-dimensional matrix.

**Algorithm De_RC**

1. Choose the least of the three kinds of words. (In this paper, we assume that modifier is the least word among vocabulary in a document)
2. The row of the matrix consists of both one kind of words and the least words chosen by previous step, such as verbs and modifiers. Hence the row can be defined as follows: \( \text{row}_j \in \{n_1 \ldots n_s\} \), \( \text{row}_k \in \{\text{adj}_1 \ldots \text{adj}_t\} \)
3. The column of the matrix consists of another kind of words and the least words chosen by step 1, such as nouns and modifiers. Then the column can be defined as follows: \( \text{col}_i \in \{v_1 \ldots v_r\} \), \( \text{col}_k \in \{\text{adj}_1 \ldots \text{adj}_t\} \)
4. The values in the matrix can be denoted as \( \text{mat}_{\text{row},\text{col}} \), then the matrix is shown in figure 1.

\[
\begin{bmatrix}
  v_1 & \ldots & v_r & \text{adj}_1 & \ldots & \text{adj}_t \\
  n_1 \\
  \vdots \\
  n_s & \text{adj}_1 \\
  \vdots \\
  \text{adj}_t
\end{bmatrix}
\]

**Figure 1.** A diagram of the matrix

5. From the matrix, we can get:

\[
P_D(n, v, \text{adj}) = \frac{\text{mat}_{\text{row}_k,\text{col}_j} \cdot \text{mat}_{\text{row}_j,\text{col}_l} \cdot \text{mat}_{\text{row}_j,\text{col}_l} \cdot 1}{\sum_i \text{count}(v_i) \sum_j \text{count}(n_j) \sum_k \text{count}(\text{adj}_k) \text{count(verbs)}}
\]

In this way, the parameters in formula (7) can be computed generally, we can achieve the generative probability for a document at a cost of additional items in both rows and columns, however vastly simplify the computational complexity from a three dimensional matrix.

### 4. Evaluating the Language Model

A Language Model is always expected to predict the k+1 word while given the first k words of a sentence. In a sense, we hope the model can give the probability distribution of the possible occurrence of the K+1 word.
Perplexity[10] is such a criterion that can be used to measure the quality of Language Models in the field of Natural Language Processing. It can estimate the probability of a sentence based on each word.

\[
\text{Perplexity(sentence)} = P(w_1 w_2 \ldots w_N)^{-\frac{1}{N}}
\]

(13)

The perplexity is algebraically equivalent to the inverse of the geometric mean per-word likelihood. A lower perplexity score indicates better generalization performance. Especially, when the language model is a unigram model, the perplexity can be:

\[
\text{Perplexity(sentence)} = 2^{-\frac{1}{N} \sum \log(P(w_i))}
\]

(14)

In this paper, \(P(v_i) = \frac{\text{count}(v_i) / \text{sentences}}{\text{count}(\text{vocabulary})}\).

The nouns and modifiers are generated as the verbs’ relevant objects or attribution, so:

\[
P(w_i) = P(v_i) \prod_{j=1} P(n_j | v_i) \prod_{k=1} P(\text{adj}_k | v_i)
\]

(15)

From formula(14) and formula(15), the perplexity score for text model which is based on linguistic analysis with different parts of speech can be obtained. Then we trained all the variables in models using the 20newsgroups[11] to compare the generalization performance with unigram, bigram, and trigram.

![Figure 2. Perplexity from text models](image)

The experiment begins at segmentation of parts of speech, the words except verbs, nouns, and modifiers are removed. Figure 2 indicates that the generative probability of a sentence proposed in this paper is approximated to the probability in a bigram when the number of words is small. However, with the increase of vocabulary, it plays a better and more accurate performance than unigram and bigram. Additionally, our text model is approximately to trigram with less relevant words involved and the changeability of the words makes the computation simpler.

Furthermore, we conduct experiments on different types of documents in the 20 newsgroup with their sizes varying from 200 to 5000 words.

From table 1, the experiment results present higher scores when the news document contains too many proper nouns and parameters, especially in the type of Motorcycles. These professional words occurred frequently in the document often represent a name for the objects’ in corresponding area, which causes very small probabilities of \(P(n_j | v_i)\) and the Perplexity grows.
Table 1. Perplexity from various documents

|                     | N=200       | N=500       | N=1000      | N=2000      | N=5000      |
|---------------------|-------------|-------------|-------------|-------------|-------------|
| Graphics            | 2(1.59197)  | 2(1.55205)  | 2(1.21148)  | 2(1.35249)  | 2(1.28547)  |
| PC.hardware         | 2(1.42305)  | 2(1.6931)   | 2(1.16733)  | 2(1.25861)  | 2(1.26879)  |
| Motorcycles         | 2(1.75798)  | 2(1.60706)  | 2(1.75008)  | 2(2.0178)   | 2(1.81253)  |
| Electronics         | 2(1.55432)  | 2(1.86683)  | 2(1.5993)   | 2(1.51268)  | 2(1.47268)  |
| Space               | 2(1.58746)  | 2(1.57852)  | 2(1.46556)  | 2(1.36985)  | 2(1.25478)  |

5. Conclusion
In this paper, we assume that corresponding words in a sentence contribute to describing certain meaning of the topic, and all kinds of description by human is fundamentally based on verbs, nouns, and modifiers. Consequently the text can be modeled according to the relationship among these three kinds of words. The generative probability of the model is given and an algorithm for dimensionality reduction is proposed to overcome the common problems of matrix built in text modeling.

In the future work, we plan to extend the text modeling to the application for document classification and feature extraction. On the other hand, the improvement for the sparse matrix in this paper is essential. More researches need to be done in the analysis and selection of semantic words.

References
[1] Salton G, Wong A, and Yang C. S. 1975 A vector space model for automatic indexing Communications of the ACM pp. 613-620
[2] Chowdhury G. 2003 Natural Language Processing Annual Review of Information Science & Technology pp.51-89
[3] Blei, David M. 2012 Probabilistic Topic Models Communications of the ACM. pp.77–84
[4] Thomas H. 2000 Learning the Similarity of Documents: an information-geometric approach to document retrieval and categorization Advances in Neural Information Processing Systems MIT Press pp.914-920
[5] Blei, David M. and Andrew Y. 2003 Latent Dirichlet Allocation, Journal of Machine Learning Research. pp.993–1022
[6] Stanford University, Computational Linguistics, Stanford Metaphysics Research Lab, Encyclopedia of Philosophy, Apr, 2017
[7] Goldberg Y. 2016 A Primer on Neural Network Models for Natural Language Processing, Journal of Artificial Intelligence Research pp.345–420
[8] Zheng J. 2017 Principle and Practice of NLP Chinese Natural Language Processing, Publishing House of Electronics Industry pp 68-71
[9] Russell, Stuart J and Norvig 2003 Artificial Intelligence: A Modern Approach, Upper Saddle River, New Jersey: Prentice Hall, 2nd edition
[10] Peter E, Vincent J 1992 An Estimate of an Upper Bound for the Entropy of English Computational Linguistics pp.31-40
[11] Xiaona W., Jia Z. and Jianfeng Y. 2014 Finding Better Topics: Features, Priors and Constraints Pacific-Asia Conference on Knowledge Discovery and Data Mining. pp. 296-310, New York, Springer