基于深度学习的旅游评论情感倾向性研究

魏东平

摘要

面对网络上日益丰富的评论信息资源，如何能快速有效的获取并使用其中的有效信息成为人们关注的问题。本论文研究目标是互联网上的旅游评论，通过使用数据挖掘算法分析获取评论中关于商品或服务的特征词，并提取所有评论中包含特征词的句子。利用 LingPipe 方法和 PMI 方法来分析这些特征句的情感倾向。并利用 LingPipe 方法和统计分析方法分析影响旅游评论有用性的因素。

本文首先系统地介绍了旅游电子商务的相关知识。通过对旅游电子商务的基本概念和分类的介绍，总结了旅游电子商务的发展现状，并提出了旅游电子商务的发展趋势，为下一步的研究奠定基础。

随后论文对文本分类和文本情感倾向分析的来源及原理作了简要地总结，介绍了几种主要的文本情感倾向分析分类的方法。在实验阶段，首先论文分析的是携程网上关于酒店的情感倾向性分析。论文本部分的研究目的是获知评论者对于某种产品或服务的某一特征的情感倾向。首先利用数据挖掘算法获得关于酒店评论者关注的特征词，并利用程序获得包含这些特征词的句子。然后论文通过使用 LingPipe 方法和 PMI 方法来分析关于某一特征的情感倾向，并根据实验过程和分类性能将两种方法进行比较。

在第二部分研究中，本文主要分析了影响互联网上旅游评论有用性的因素。搜集整理了 yahoo 英文中旅游频道上的关于旅游目的地的评论，利用 LingPipe 方法获得评论中的主、客观特征值，建立固定效应对数线形回归方程和其改进模型分析得出主观、客观相交融且比较长的评论感知有用性最强。

本研究可以快速地从海量评论中获得评论者对于某一产品或服务的某一特征的正负面评价，能更为有效地辅助阅读者的决策。且可以了解什么样的评论对阅读者帮助最大，相信本研究在未来能够得到很好的实践应用。

第 1 章 绪 论

1.1 研究背景及意义

21 世纪以来，随着计算机软硬件及相关技术的飞速发展和计算机网络的日新月异，Internet 普及程度越来越高，标志着我们已经全面进入了信息时代。通过 Internet 进行信息检索不仅仅是政府、公司、集团获取信息的重要手段，而且也为个人的学习、生活和工作带来了便捷。在 Internet 上，有数十亿的网页，成千上万 TB 的数据，包括文本、图像、声音、影像，这些类别又分别以各式各样的格式存在，包括 HTML、XML、Po F、JPEG、MPEG、MP3 等等。另外，每天有数十万的网页更新，数百万的新的网页加入，使得 Internet 上的信息丰富而又复杂。人们需要的绝大部分信息 Internet 都能提供。随着经济和技术的飞速发展，电子商务的出现，网络经济发展到了一个新的高度。B2C、C2C 等商务模式促使通过网络进行交易的数量大幅增加。第 20 次中国互联网络发展状况统计报告表明，用户 25.5%的网络活动为利用互联网进行网络购物。

但是与传统交易相比，消费者选择网上购物面临无法在购物前实际看到商品，从而无法对于商品质量、品质等有客观的认识；与商家没有面对面的交流；没有实际中的物理店铺的出现。消费者对网上的信息并不完全信任，而且担心付款后收不到商品或收到的商品与预想的有差距。所以对于网络购物来说，商品或商家的相关评论就显得尤为重要。研究表明，消费者之间交换的信息会影响到其他消费者的消费选择。来自于亲友或朋友的意见经常被认为是最重要的消费前的信息。而 Internet 技术与应用在过去十几年中的快速发展使得互联网

已经成为人们最重要的信息来源之一。人们获取信息的来源不再只局限于周围的亲友，而是扩大到了更加广泛的范围。“网络信息”不仅指网络媒体的各种报道，也包括发布在网上论坛及博客中的帖子、留言等，每个人都可以通过“发帖子”的方式把自己的经历、思想和观点公诸于众。

2005 年，DoubleClick 公司进行的调查表明，50%以上的买家在进行网络购物时会去网上搜索相关信息来支持自己的消费决定。在四种不同类型的产品和服务中，旅游消费者进行消费前信息搜索的比率最高，占 73%（如图 1-1）。

以 Internet 为代表的信息技术推动了旅游业的第三次革命。第 20 次中国互联网络发展状况统计报告表明，旅游者出行全部通过网上预订的为 7.4%，而通过网络进行部分预订或是查询信息的为 90.2%，只有 2.3%的旅游者即不在网上预订，也不在网上查询信息。

旅游业被公认为发展电子商务最为得天独厚的行业，因为旅游相对于其它产业而言，其产品具有无形性、不可储藏性和信息供需量大的特点，它几乎不需要物流配送环节，其产品的销售过程既是信息(食、住、行、购物、娱乐、学习等)的传递过程，也是旅游服务过程。所以，旅游企业需要电子商务加速自身的发展，电子商务也可以通过旅游企业的应用来释放其巨大的能量。

电子商务运用于旅游业虽然只是近几年的事，但其发展速度却是惊人的。据 CNN 的数据显示[1]：“1999 年度全球电子商务销售额突破 1400 亿美元，其中旅游业电子商务销售额突破 27 亿美元，占全球电子商务销售额的 20%以上。全球约有超过 17 万家的旅游企业在网上开展综合、专业、特色的旅游服务，约有 8500 万以上人次享受过旅游网站的服务，全球旅游电子商务连续五年以 350%以上的速度发展。”就目前全球范围看，电子商务已成为旅游业开拓新市场，研发、创新旅游产品和旅游企业树立品牌形象、提高竞争能力的全新工具和媒介手段。

目前我国的旅游网站主要有四类[2]：一为门户网站（如新浪、搜狐、网易等）的旅游频道，对一些旅游景点、旅游路线、旅游知识等方面的作简单地介绍，但因不能全面地提供整套的旅游服务和缺乏专业资源的支持而竞争力不强；二为传统旅游企业自建的网站，如昆仑在线、青旅在线等。这类网站虽基本具备宣传自我和发布信息的功能，但还不能提供全面的旅游服务；三为专业旅游网站，如携程网、中国旅游资讯网、西部旅游信息网等，它们一般有风险投资的背景，能以良好的个性服务和强大的交互功迅速抢占网上的旅游市场份额；四为政府背景类 ASP（应用服务供应商）网站。这类网站以国家“金旅工程”公共商务网———金旅雅途网为代表，它们自身不经营旅游业务，而是要构建一座“旅游电子商厦”，为旅游企业提供电子商务平台,以全面解决旅游企业上网营销、管理的问题。

web2.0 技术的出现，对于旅游电子商务的发展提出了新的发展机会和挑战，通过 Internet 可以实现旅游产品供应商与消费者的双向交流及消费者之间的双向交流。这使得网上充满了海量的关于旅游产品或目的地的信息和评论。旅游相关的评论会出现在很多地方，旅游论坛，旅友博客，或者是订购旅游相关服务时会有已使用过相关服务的旅游者的评价等等。随着网络上旅游相关的信息的增多，人们对于去什么地方旅游，去旅游目的地使用什么相关服务有了越来越多的选择；对于旅游业经营者而言，这些在线留言会让他们认清自己在市场中的位置。而这些网络中未处理过的信息是海量的，如在 Google 中搜索“香港旅游留言”，Google 会返回超过 4 百万个相匹配的网页，几乎没有人能阅读所有这些留言。而现在的各种旅游网站并没有提供信息分类和处理的功能，这使得阅读者面临无法识别有用评论和快速获得信息的困难。

本文将利用数据挖掘技术对评论信息进行处理，帮助阅读者快速从海量信息中获得需要的信息，对于旅游业经营者来说，可以帮助其认清竞争对手的情况和消费者对于其提供的产品和服务的态度，帮助经营者调整经营策略，同时对其旅游电子商务的发展具有一定的可借

鉴意义。

1.2 国内外研究现状

1.2.1 基于浅层学习的短文本情感分析研究现状

1.2.2 基于深度学习的短文本情感分析研究现状

1.3 本报告主要研究工作及内容安排

第 2 章 自然语言模型及其改进

In the contemporary world internet applications such as Micro-blog, Weibo,Wechat , Facebook, and Twitter etc,. have become the main platforms for people to express their views and feelings or sentiments. Huge rise in the number of internet users has resulted in proportional rise in the volume of their relevant messages. The ways to extract the current social trends of public opinion from the social media that could be utilized by the Governments and enterprises to make timely and effective decision-making, is becoming a hot topic in the field of natural language processing [1]. Natural language processing (NLP) is using computer to process and understand large scale text in internet. In fact, NLP is an important topic of the field of artificial intelligence. The traditional statistical language models include N-gram Model and N-pos Model, Maximum Entropy Model, and some others.

In recent years machine learning model has become the mainstream of natural language processing research [2]. In previous ten years, most of the machine learning methods in the field of NLP belongs to the shallow layer learning model. The common shallow layer learning model includes the classification model based on support vector machine (SVM), the sequential tagging model based on conditional random field, logistic regression, Naive Bayes and others [3]. The shallow models almost require artificial experience to extract features from the text. The machine learning model is mainly responsible for classification or prediction. The performance of the models is often determined by the quality of features extracted artificially. Therefore, researchers have to conduct long studies to observe and extract the features of the text. Researcher needs to extract the features for different tasks. Relatively the more difficult problem in extracting effective features is the requirement from the researcher to have a rich experience and comprehensive understanding of the data.

In Natural language processing, the commonly used word vector representation methods are One-hot Representation (Bag-of-word), Distributed Representation. In previous studies, the One-hot Representation of words is often found simple and rough [4]. A word is generally expressed as a high-dimensional vector consistent with the size of the dictionary. Each position of the high-dimensional vector corresponds to a particular word in the dictionary. A particular word in corresponding position is 1, otherwise 0. There are two problems in One-hot representation method. Firstly, it has a very high dimension and is extremely sparse. Its high dimension and sparsity can cause the common "dimensionality disaster" in natural language processing. For example, a vector with a dimension of 100,000 is entered into a neural network model. Under this dimension, the computational cost is very high even for simple applications when using neural network model. However, it cannot represent complex semantic information in a natural language such as the similarity between words and words. This method also makes it impossible to calculate the basic semantic correlations in the language.

In recent years, a large number of researchers have a clear inclination towards Distributed Representation express the text. Bengio uses the neural network model to get a vector representation of word embedding (word vector) [5]. The word vector is a low dimensional, dense, continuous vector representation. In addition, the word vector can also contain both semantic and syntactic information in the text. Word vectors not only effectively avoid the problem of dimensionality disaster and sparsity, but also the semantic correlation between words can be computed easily. In the result of Bengio's study, most natural language processing methods under neural networks are based on the input of word vectors. The researchers improved the word vector under the different deep learning structure such as the RNN(recursive neural network) or the recursive automatic encoding (recursive auto-encoder), CNN(Convolutional Neural Networks) [6], etc. The current method leads basically to the learning of word vectors in large scale data in an unmonitored way, ignoring the characteristics of human language learning and some very important inherent attributes of language which should be collected artificially.

We will review the traditional statistical language models such as N-gram model, N-POS model, the maximum entropy model including some others to find their principles, weakness and advantages in the following sections. More over we will also review popular word vector model such as traditional one-hot representation, Bengio's word vector, Mikolov's CBOW, and Skip-gram to analyze these model structures for their advantage and weak points that could be improved. Subject to these precise reviews, we propose our new model PSWV-model to improve the performance of word vector in NLP Task.

2.2.1 基于统计学的自然语言模型

2.1 Statistical language models

In the traditional statistical language model, the goal of the model is to compute the probability of each sentence appearing in the corpus. The conditional probability of each sentence appearing is finally obtained by multiplying the conditional probability of each word in a sentence:

$$p(s) = p(w^T) = \Pi_{t=1}^{T} p(w_t|context) \tag{2.1}$$

Where $\mathrm{w}_1^{\mathrm{T}} = (\mathrm{w}_1, \mathrm{w}_2, \cdots \mathrm{w}_{\mathrm{T}})$ and w_{i} is the i-th word in the sentence. According to the different partitioning methods of context, it can be divided into different language models. Model (2.1) is defined as unigram model when $\mathrm{context} = \mathrm{null}$. When $\mathrm{context} = \mathrm{w}_{\mathrm{t-n+1}}, \mathrm{w}_{\mathrm{t-n+2,\cdots,}}\mathrm{w}_{\mathrm{t-1}}$ model (2.1) is proposed as N-gram model. The objective of N-gram model optimization is the maximum log likelihood as following

$$\Pi_{t=1}^{T} p_t(w_t|w_{t-n+1}, w_{t-n+2,\cdots,} w_{t-1}) log\ p_t(w_t|w_{t-n+1}, w_{t-n+2,\cdots,} w_{t-1}) \quad (2.2)$$

when context = $\mathrm{w}_{t-3}, \mathrm{w}_{t-2,\cdots,} \mathrm{w}_{t-1}$ model (2.1) is trigram model.

N-POS model is a derivation of N-gram model. It assumed that the probability of t-th depend on the $\mathrm{n}-1$ words' part-of-speech POS . N-POS model can be expressed as following:

$$p(s) = p(w^T) = \Pi_{t=1}^{T} p_t(w_t|c(w_{t-n+1}), c(w_{t-n+2)}, \cdots, c(w_{t-1)}) \quad (2.3)$$

where $\mathrm{c}(.)$ is mapping function of part of speech.

In natural language processing, another well known model is the maximum entropy model. It is proposed that the event is predicted by the full information of the event and no assumptions are made on the predicted event. The prediction can reduce the risk and obtains the maximum entropy value. The maximum entropy model is shown as following

$$p(w|context) = \frac{e^{\sum_i \alpha_i f_i(context,w)}}{z(context)} \quad (2.4)$$

where $\mathrm{z(context)}$ is the is the expectation of all words appearing in the context of word w.

2.2.2 基于神经网络语言模型

2.2 The Neural network language model

In Natural language processing, the commonly used word vectors is one-hot representation and Distributed representation. The idea of One-hot representation is to express words into a very long vector. The dimension of the vector is the size of the lexicon. This representation method possesses two challenging shortcomings: 1) dimensionality disaster: the dimension of word vector usually reaches tens of thousands of dimensions, hence the word vector is difficult to apply in deep learning; 2) lexical gap phenomenon: the similarity between words cannot be described.

In 1986, Hinton proposed Distributed Representation to solve the above two problems. Bengion (2002) further proposed a neural network language model and its structure shown in the following Fig.2.1.

In Fig.2.1, $w_{t-n}, \ldots, w_{t-2}, w_{t-1}$ indicate the $n-1$ words before the tth in the context or sentence, the word w is mapped $C(w)$ to a vector firstly. C is a $|V| \times m$ matrix. $|V|$ represents the size of the lexicon. m represents the dimension of a word vector, usually set to several hundred dimensions.

The first layer of the neural network is the input layer. The calculated method in the input layer is concatenating vectors. The word vectors are connected in sequence, and then $(n-1) \times m$ vectors of are obtained as following

$$X = C(C(w_{t-n}), \cdots, C(w_{t-2}), C(w_{t-1}))$$

The second layer is the hidden layer, and the vector X is transformed linearly as following

$$HX + d \quad (2.5)$$

where H is the weight matrix of X and d is bias. Then nonlinear transformation results as the active function, and then the result of transformation function is used as the input of the third layer.

$$tanh(HX + d) \quad (2.6)$$

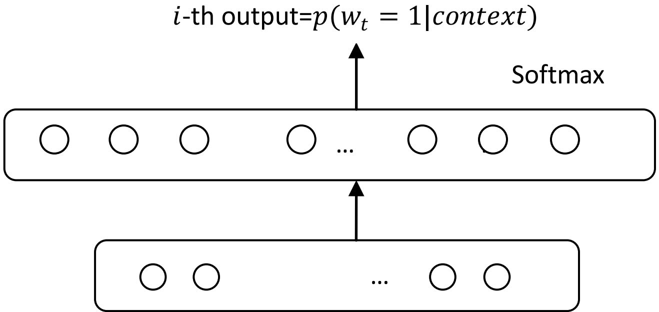

Fig.2.1 The structure of neural network language model

The third layer is the output layer. The output layer contains $|V|$ neurons. The value of neurons y_t under the condition of the context $= \{w_{t-n}, \ldots, w_{t-2}, w_{t-1}\}$ is the un normalized log probability of the tth word. The probability distribution of y in the output layer after transformed by the input layer and hidden layer is as follows

$$y = b + \omega x + Utanh(HX + d) \tag{2.7}$$

where ω, U, H is the weight matrix between layers of the neural network. And b，d is the bias term. Finally, y is normalized by using softmax function and the constraints of the probability distribution of y are as follows:

$$p(w_t|w_{t-n}, \ldots, w_{t-2}, w_{t-1}) = \frac{e^{y_{w_t}}}{\sum_i e^{y_{w_i}}} \tag{2.8}$$

Parameters $\theta = (\omega, U, H, d, b, C)$ updating in the model adopts SGA(stochastic gradient ascent) as following

$$\theta \leftarrow \theta + \eta \frac{\partial logp(w_t|w_{t-n}, \ldots, w_{t-2}, w_{t-1})}{\partial \theta} \tag{2.9}$$

where η is the learning rate, that is the step size of updating when the gradient rising in the process of model solving.

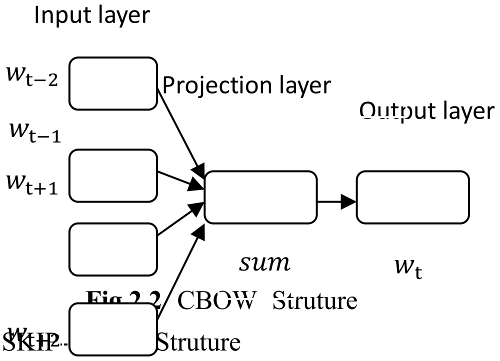

Fig.2.2 CBOW Struture

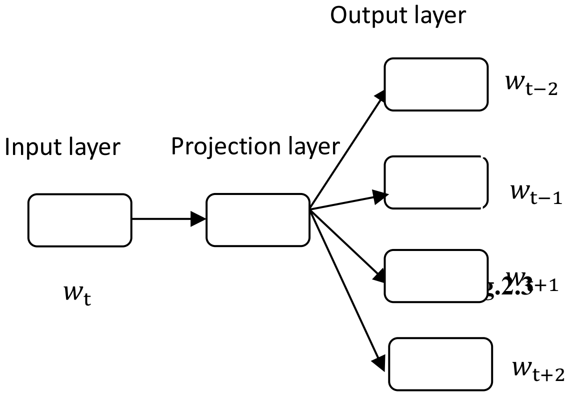

Fig.2.3 SKIP Struture

Based on the neural network language model of Bengio, Mikolov proposed two model including CBOW(Continuous Bag-of-words) and Skip-Gram in 2013 [7].CBOW use the context of the given word w to find the probability of the word w, while Skip-Gram is the probability of the context of the known word w. CBOW and the Skip-Gram model framework are shown in the following Fig.2.2 and Fig.2.3. It consists of input layer, projection layer and output layer. The biggest difference between

CBOW and Bengio's neural network language model is that the hidden layer is removed because of its huge amount of calculation and the projection layer is added. In order to further reduce the computational complexity, Mikolov changed concatenating vectors in the hidden layer into calculating the sum of vectors in the projection layer.

The learning objective function of CBOW is to provide a maximum logarithmic likelihood function according to the context (w_{t-2}， w_{t-1}, w_{t+1}, w_{t+2}) when the learning window is $[-2,2]$.

$$L = \sum_{w \in C} logp(w|context(w)) \quad (2.10)$$

where w is any word in corpus C, and the number or dimension of vocabularies in the corpus is |C|. The above problem can be transformed into a multi-classification problem. The commonly used method is the softmax regression. However, softmax regression needs to compute the probability and normalization of the |C| vocabularies. The amount of computation is large, so Mikolov puts forward two improvements: adopting hierarchical softmax and negative sampling.

2.3 基于深度学习词向量模型

2.3.1 CBOW 模型

2.3.2 Skip-Gram 模型.

2.3.3 Glove 模型

基于词性及情感极性的深度学习语言模型

3 The Proposed model

The reviews of neural network language model and its improvements reveal that these models can remove dimensionality disaster and lexical gap phenomenon in contrast with traditional statistical language model. Especially, the Mikolov’s improvements simplify the neural network structure and reduce the calculation complexity. However, these improvements drop some features such as the characteristics of part of speech and the sentiment information of words inherent in the natural language. In following sections, we will focus on maintaining the word order of the context, and take the characteristics of part of speech and the sentiment information of words into the consideration of the model under the basis of the neural network model and its improvement.

3.1 Preserving word order of the context

In order to reduce the time complexity of the model, the CBOW proposed by Mikolov in 2013 changed the vector concatenating into vector summing in the input layer of Hinton's neural network language model in 2002, and used softmax in the output layer. But this will sacrifice the word order feature of context. We think that under the increasing computation speed, we can keep the word order feature to capture more semantic information by keeping the concatenating vectors proposed by Hinton in the input layer.

3.2 The characteristics of part of speech

At present, most word vector models seldom use grammatical knowledge. However, it is well known that grammatical knowledge is very important for understanding of the language. And the semantics and usage of the same word is different when the part of speech is different. For example, the English word “like” means “prefer or wish to do something” ,when it is a verb. It means “having the same or similar characteristics” when it is an adjective. Obviously, the part of speech is important to understand the words.

Scholar Baotian Hu published an article “A novel word embedding learning model using the dissociation” [8] in Neurocomputing in 2016. He proposed the CDNV (continuous dissociation between nouns and verbs model) with the ideas to categorize words as verbs, nouns and others when developing the word vector model. Hu proposed three methods to separate the words with parts of speech, including CDNV-1, CDNV-2, CDNV-3, and the maximum separation degree is CDNV-3.

In fact, part of speech can be divided into many kinds, such as verbs, nouns, prepositions, adjectives, adverbs, etc. Firstly, we use the language tagging tools to annotate the corpus to obtain the training set which contains the feature of parts of speech. The categories of POS tagger may include VB/NN/CD/JJ/MD/IN/CC/RB/PRP etc. In our model, the same word may also be defined as a different encoding. $\mathrm{C(w_i, p_i)}$ is the encoding of a word $\mathrm{w_i}$ when part of speech of the word is $\mathrm{p_i}$,where $\mathrm{p_i} \in \{\mathrm{VB, NN, CD, JJ, RP, IN, CC, RB, PRP, PBR}\}$.

3.3 The sentiment information of words

The word vector models of Bengio and Mikolov can solve the two words’ semantic isolation problems in one-hot representation, which can make word vector of the two words with similar semantic close to each other. Thus their word vector models can make many NLP tasks, such as clustering part of speech analysis, synonym analysis, and the others, more clear and simple. However, Bengio and Mikolov's word vector models and its subsequent improved models ignore a key information in natural language, that is the sentiment information of words. Some words, such as “beautiful” and “ugly”, “love” and “hate”, have the opposite sentiment polarity. In that case, the distance of the word vectors should be far away from each other. On the other hand, the distance in the expression of word vector of the similar sentiment polarity words such as “smile” and “laugh” should be closer to each other.

Andrew L.Mass 's (2011) paper “learning word vector for sentiment analysis” [9] proved that words with opposite sentiment polarities have closer distance in current word vector model because grammatical and usage similarities are closer to each other in the semantic space. Such word vectors often remain unsatisfactory in the task of sentiment analysis. Andrew L.Mass suggested that it is better to consider both semantic and emotional information of words in training word vectors procedures. Andrew L.Mass also proposed that the probability of whole samples can be obtained by unsupervised continuous probability distribution model to catch the semantic information of context. And then he optimized the word vector through a supervised model by introducing annotated context. Finally, the two parts are combined together to form the objective function, so that his word vector contains not only semantic information, but also emotional information.

In this paper, we will consider the unification of semantic information and sentiment information in the framework of CBOW model proposed by Mikolov. We shall attempt to improve CBOW model to obtain the semantic information and sentiment information of the context. We shall pursue a supervised context for sentiment polarity labeling in our model. The emotion of texts is divided into subjective and objective. The objective text does not contain any emotion classification information. Subjective text contains emotional information, such as love, dislike, and sadness, etc,. Normally it divides emotion of text into two, three or multi categories. The two classifications are divided into positive and negative. Three classifications are divided into positive, negative, and neutral. The multi sentiment polarity of text is commonly classified in the following categories: joy, anger, sadness, shock, and fear. In addition, OCC-model proposes that emotional cognition can be divided into three categories and 21 subgroups. For example, Qiao Xiangjie [10] (2010) proposed that e-learning students have eight emotions based on OCC-model: pleasure / sadness, satisfaction / disappointment, gratitude / anger, shame / pride. The sentiment intensity of text can be divided into different step-type grades such as like, like very much, annoying, hate. The other way is trying to define the sentiment intensity of text by scoring 1, 2, 3, 4, and 5. We divide the sentiment polarity $\mathrm{s_w}$ of word w into n-grades such as $s_w \in \{s_1^w, s_2^w, \cdots, s_n^w\}$.The sentiment polarity is separated by Huffman tree under the framework of CBOW model. The typical separation structures are shown in the figures as below, where $s_P^w = \{s_1^w, s_2^w, s_3^w\}$,$s_N^w$ =$\{s_4^w, s_5^w\}$.

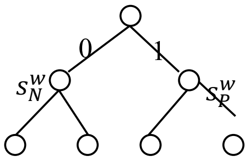

Fig3.1 the positive and negative separation structure

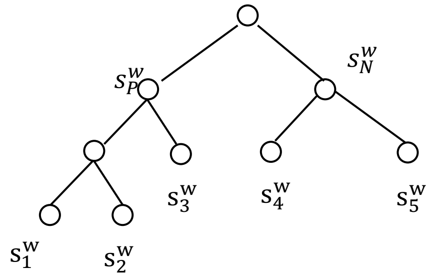

Fig3.2 the five grades separation structure

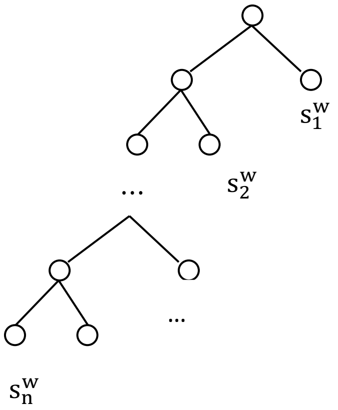

Fig3.3 the multi-grades separation structure

3.4 Word Vector Model based on part of speech and sentiment Information（PSWV-model）

We conclude by proposing a model to hold the word order features, part of speech features and sentiment polarity feature of text on the basis of CBOW model. We shall focus on obtaining more semantic and emotional information of texts by the training of our word vector model. We shall define our word vector model as PSWV-model based on part of speech and sentiment information. There are three layers in our model, such as input layer, projection layer and output layer, as shown in Fig.3.4. The training window size of the word w is n. For example, if $n = 2$, then the probability of the word w is obtained by the information of the two words in front of, and the next two words of it.

Given current word is $\mathrm{w_t}$, we take four rows in the look up table or word vector Matrix C as the input word vector of w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2}. C is a matrix and its size is $|V| \times m$. $|V|$ is the size of the word table. Super parameters m is the word vector's dimension and needs to be set by the users. m is usually set to a few hundred dimensions. The first layer of the model is to map the word vectors from the look up table C and connect the context of word $\mathrm{w_t}$ orderly.

$$X_{nm} = (C(w_{t-2}), C(w_{t-1}), C(w_{t+1}), C(w_{t+2})) \qquad (3.1)$$

However, Mikolov's CBOW model calculate the sum of the word vectors $\mathrm{w_{t-2}}$, $\mathrm{w_{t-1}}$,

w_{t+1}, w_{t+2}. In that case, it ignores word order information when summing the word vectors. Our PSVW-model concatenate the word vectors of w_{t-2}, w_{t-1}, w_{t+1}, w_{t+2} .Thus we can obtain word order information. Mikolov's CBOW model simplifies the complicated computation by removing the hidden layer of Bengio's neural network language model in order to reduce the computational complexity quite significantly. Our PSVW-model absorbs the advantage of CBOW model. The hidden layer is simplified to projection layer, whose function is to transfer the word vector information from the input layer to the output layer.

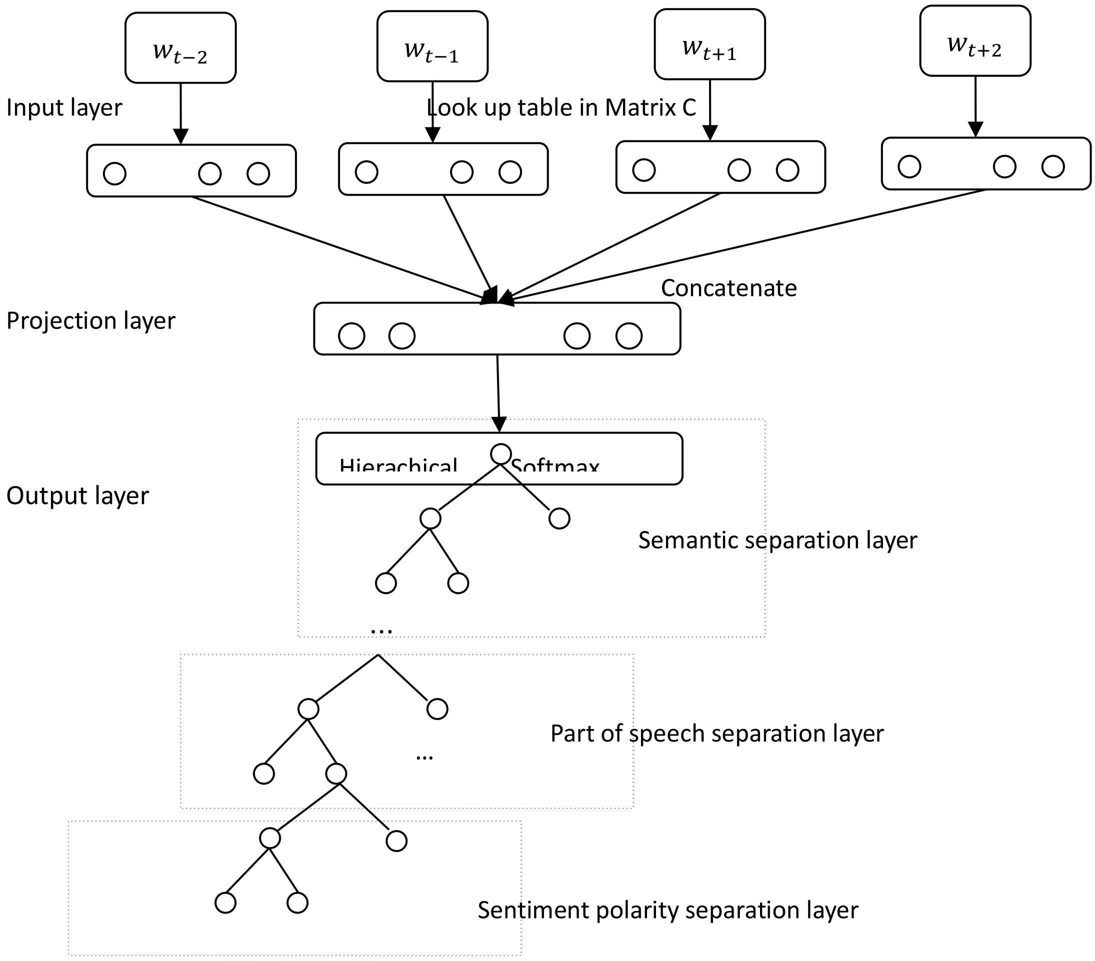
w_{t-2}
w_{t-1}
w_{t+1}
w_{t+2}
Input layer
Look up table in Matrix C
Concatenate
Projection layer
Hierarchical Softmax
Output layer
Semantic separation layer
...
Part of speech separation layer
...
Sentiment polarity separation layer

Fig.3.4 the structure of PSWV-model

The third layer is the output layer. The neural network language model used the traditional softmax classifier in the output layer. The number of nodes in the output layer is the size of the dictionary. The size of the parameter matrix is $m \times |V|$. Then the output layer has a huge amount of computation in output layer of the neural network language model. Hence, Mikolov tried to use the hierarchical softmax optimization mechanism in the output layer. Actually, the computational complexity can be significantly reduced by using hierarchical softmax algorithm. We can also use hierarchical softmax algorithm in our PSWV-model. For example, we try to model the sentence " I like to read love stories". Assuming that the predictor word is "like" and the learning window is $n = 2$.Then the preceding word is "I" and the following two words is "to read". Since there is only one word above, the missing word can be replaced by a placeholder *PADDING*.The words in learning window are mapped into word vectors in matrix C, respectively. Then we can concatenate the four word vectors orderly and transmit to projection layer. After that information of part of speech and its sentiment polarity of the word are transferred together to hierarchical softmax classification in output layer. Finally, the word vector model can fuse with the information of word order, the sentiment polarity of words and part of speech (POS).

Shown in Fig3.4, we adopt the hierarchical softmax algorithm in the output layer. Then, the output layer becomes a Huffman tree. The Huffman tree consists of three layers: the first layer is the semantic separation layer; the second layer is the POS separation layer; the third layers is sentiment polarity separation layer. The non-leaf node of Huffman tree represents a two-class operation. If the output is 0, then the path goes left. If the output is 1, then the path goes right. Passing through the semantic separation layer, the POS separation layer and sentiment polarity separation layer, each leaf node represents a word in the corpus with a unique POS feature and sentiment polarity feature. Therefore, each word in Huffman tree may have different codes when the word has different POS and sentiment polarity features. The probability of each word in the corpus with different POS feature and sentiment polarity feature is the product of the probability of the root node and all the non-leaf nodes in the path of root to leaf. Hence, our PSWV-model should maximize the logarithmic likelihood function as follows:

$$L = \sum_{w_t \in C} logP(w_t | {W_t}^T, POS, SP) \qquad (3.2)$$

where ${W_t}^T$ =(w_{t-n}, w_{t-n+1}..., w_{t+2}, $w_{t+n})^T$and POS is the part of speech of the word $\mathrm{W_t}$. SP is the sentiment polarity feature of the word $\mathrm{W_t}$.

In order to describe PSWV-model clearly, we define the following symbols.

R^w represents the path from the root node of Huffman tree to the leaf node corresponding to the word W.

R^c is the path from the root node of Huffman tree to the semantic separation layer.

R^p is the path of the word w in the POS separation layer.

R^s is the path of the word w in sentiment polarity separation layer .

n^w is the number of nodes passed from the root node of Huffman tree to the leaf node corresponding to the word W.

n^c is the number of nodes passes through the semantic separation layer corresponding to the word W.

n^p is the number of nodes passes through the POS separation layer corresponding to the word W.

n^s is the number of nodes passes through the sentiment polarity separation layer corresponding to the word W.

where $n^w = n^c + n^p + n^s$.

$P_1^w, P_2^w, \cdots, P_{n^c}^w$ are the nodes of the word W passes over the semantic separation layer.

$\hat{P}_1^w, \hat{P}_2^w, \cdots, \hat{P}_{n^p}^w$ are the nodes of the word W passes over the POS separation layer.

$\tilde{P}_1^w, \tilde{P}_2^w, \cdots, \tilde{P}_{n^s}^w$ are the nodes of the word W passes over the sentiment polarity separation layer.

where $P_{n^c}^w = \hat{P}_1^w$ represents the common transformation node between the semantic separation layer and the POS separation layer.

Where $\hat{P}_{n^p}^w = \tilde{P}_1^w$represents the common transformation node between the POS separation layer and sentiment polarity separation layer.

$\xi_1^w, \xi_2^w, \cdots, \xi_{n^c}^w$ are the Huffman encoding of the nodes of the word W passes over the semantic separation layer, where ξ_j^w is the code of P_j^w.

$\hat{\xi}_1^w, \hat{\xi}_2^w, \cdots, \hat{\xi}_{n^p}^w$ are the Huffman encoding of the nodes of the word W passes over the POS separation layer, where $\hat{\xi}_j^w$ is the code of the $\hat{P}_j^w$.

$\tilde{\xi}_1^w, \tilde{\xi}_2^w, \cdots, \tilde{\xi}_{n^s}^w$ are the Huffman encoding of the nodes of the word W passes over the sentiment polarity separation layer, where $\tilde{\xi}_j^w$ is the code of the $\tilde{P}_j^w$.

$\theta_1^w, \theta_2^w, \cdots, \theta_{n^c}^w$ are the parameter vector of non-leaf node in the path R^c .

$\hat{\theta}_1^w, \hat{\theta}_2^w, \cdots, \hat{\theta}_{n^p}^w$ are the parameter vector of non-leaf node in the path R^p.

$\tilde{\theta}_1^w, \tilde{\theta}_2^w, \cdots, \tilde{\theta}_{n^c}^w$are the parameter vector of non-leaf node in the path R^s.

With these definitions, we can convert the conditional probability in Eq.(3.2) into:

$$P\left(w_t \middle| W_t{}^T\right) = \prod_{i=2}^{n^c} p(\xi_i^w \,|X_w, \theta_{i-1}^w) \prod_{j=2}^{n^p} p(\hat{\xi}_j^w \,|X_w, \hat{\theta}_{j-1}^w) \prod_{k=2}^{n^s} p(\tilde{\xi}_k^w \,|X_w, \hat{\theta}_{k-1}^w) \tag{3.3}$$

where

$$p(\xi_i^w|X_w, \theta_{i-1}^w) = \begin{cases} \rho\left(X_w{}^T\theta_{i-1}^w\right) & \xi_i^w = 1 \\ 1-\rho\left(X_w{}^T\theta_{i-1}^w\right) & \xi_i^w = 0 \end{cases} \tag{3.4}$$

$$p\left(\hat{\xi}_j^w \middle| X_w, \hat{\theta}_{j-1}^w\right) = \begin{cases} \rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right) & \hat{\xi}_j^w = 1 \\ 1-\rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right) & \hat{\xi}_j^w = 0 \end{cases} \tag{3.5}$$

$$p\left(\tilde{\xi}_k^w \middle| X_w, \tilde{\theta}_{k-1}^w\right) = \begin{cases} \rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right) & \tilde{\xi}_k^w = 1 \\ 1-\rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right) & \tilde{\xi}_k^w = 0 \end{cases} \tag{3.6}$$

$\rho(\cdot)$ is sigmoid function. Hence, $p(\xi_i^w|X_w, \theta_{i-1}^w)$ in the semantic separation layer can be simplified as following

$$p(\xi_i^w|X_w, \theta_{i-1}^w) = [\rho\left(X_w{}^T\theta_{i-1}^w\right)]^{1-\xi_i^w}[1-\rho\left(X_w{}^T\theta_{i-1}^w\right)]^{\xi_i^w} \tag{3.7}$$

In the POS separation layer, $p\left(\hat{\xi}_j^w \middle| X_w, \hat{\theta}_{j-1}^w\right)$ can be simplified as following

$$p\left(\xi_j^w \middle| X_w, \hat{\theta}_{j-1}^w\right) = [\rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right)]^{1-\hat{\xi}_j^w}[1-\rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right)]^{\hat{\xi}_j^w} \tag{3.8}$$

In the sentiment polarity separation layer, $p\left(\tilde{\xi}_j^w \middle| X_w, \tilde{\theta}_{j-1}^w\right)$ can be simplified as following

$$p\left(\tilde{\xi}_k^w \middle| X_w, \tilde{\theta}_{k-1}^w\right) = [\rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right)]^{1-\tilde{\xi}_k^w}[1-\rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right)]^{\tilde{\xi}_k^w} \tag{3.9}$$

Hence, Eq.(3.2) can be converted as following

$$L = \sum_{w_t \in C} log \prod_{i=2}^{n^c} p(\xi_i^w \,|X_w, \theta_{i-1}^w) \prod_{j=1}^{n^p} p(\hat{\xi}_j^w \,|X_w, \hat{\theta}_{j-1}^w) \prod_{k=1}^{n^s} p(\tilde{\xi}_k^w \,|X_w, \tilde{\theta}_{k-1}^w) \tag{3.10}$$

Substituted Eq. (3.6)-(3.9) to (3.10), then Eq.(3.3) can be converted as following

$$L = \sum_{w_t \in C} \{ \sum_{i=2}^{n^c} \{(1-\xi_i^w) log\rho\left(X_w{}^T\theta_{i-1}^w\right) + \xi_i^w log\,[1-\rho\left(X_w{}^T\theta_{i-1}^w\right)]\} +$$

$$\sum_{j=2}^{n^p} \{(1-\hat{\xi}_j^w) log\,[1-\rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right)] + \hat{\xi}_j^w log\,[1-$$

$$\rho\left(X_w{}^T\hat{\theta}_{j-1}^w\right)]\}+$$

$$\sum_{k=2}^{n^s} \{(1-\tilde{\xi}_k^w) log\,[1-\rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right)] + \tilde{\xi}_k^w log\,[1-\rho\left(X_w{}^T\tilde{\theta}_{k-1}^w\right)]\}\} \tag{3.11}$$

In logarithmic likelihood function (3.11), the parameters include: θ_{i-1}^w, $\hat{\theta}_{j-1}^w$, $\tilde{\theta}_{k-1}^w$, X_w, where θ_{i-1}^w is the parameter vector of non-leaf nodes of the semantic separation layer. $\hat{\theta}_{j-1}^w$ is the parameter vector of nodes in the POS separation layer. $\tilde{\theta}_{k-1}^w$ is the parameter vector of sentiment polarity separation layer. X_w is concatenating vetors $w_{t-n}, w_{t-n+1} \ldots, w_{t+2}, w_{t+n}$ orderly.

According to optimization theory, in order to maximize the likelihood function , we must obtain the partial derivative of the parameters θ_{i-1}^w, $\hat{\theta}_{j-1}^w$, $\tilde{\theta}_{k-1}^w$, X_w

$$\frac{\partial L\,(\theta_{i-1}^w,\ \hat{\theta}_{j-1}^w,\ \tilde{\theta}_{k-1}^w,\ X_w)}{\partial\theta_{i-1}^w} = \frac{\partial(1-\xi_i^w) log\rho\left(X_w{}^T\theta_{i-1}^w\right) + +\xi_i^w log\,[1-\rho\left(X_w{}^T\theta_{i-1}^w\right)]}{\partial\theta_{i-1}^w} \tag{3.12}$$

According to the features of sigmoid function $\rho'(x) = \rho(x)(1-\rho(x))$, we can obtain Eq.(3.13) from Eq.(3.12).

$$\frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \theta_{i-1}^{w}} = [1 - \xi_i^w - \rho(X_w{}^T \theta_{i-1}^{w})]X_w \qquad (3.13)$$

Similarly, we can obtain the following Eq.(3.14) to Eq.(3.16)

$$\frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \hat{\theta}_{j-1}^{w}} = [1 - \hat{\xi}_i^w - \rho(X_w{}^T \hat{\theta}_{j-1}^{w})]X_w \qquad (3.14)$$

$$\frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \tilde{\theta}_{k-1}^{w}} = [1 - \tilde{\xi}_i^w - \rho(X_w{}^T \tilde{\theta}_{k-1}^{w})]X_w \qquad (3.15)$$

$$\frac{\partial L\left(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w\right)}{\partial X_w} = [1 - \xi_i^w - \rho(X_w{}^T \theta_{i-1}^{w})]\theta_{i-1}^{w}$$
$$+[1 - \hat{\xi}_i^w - \rho(X_w{}^T \hat{\theta}_{j-1}^{w})]\hat{\theta}_{j-1}^{w} + [1 - \tilde{\xi}_i^w - \rho(X_w{}^T \tilde{\theta}_{k-1}^{w})]\tilde{\theta}_{k-1}^{w} \qquad (3.16)$$

In the procedures of parameters training, we adopt stochastic gradient ascending (SGA) to update the parameters.

$$\theta_{i-1}^{w} \leftarrow \theta_{i-1}^{w} + \eta_1 \frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \theta_{i-1}^{w}} \qquad (3.17)$$

$$\hat{\theta}_{j-1}^{w} \leftarrow \hat{\theta}_{j-1}^{w} + \eta_2 \frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \hat{\theta}_{j-1}^{w}} \qquad (3.18)$$

$$\tilde{\theta}_{k-1}^{w} \leftarrow \tilde{\theta}_{k-1}^{w} + \eta_3 \frac{\partial L(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w)}{\partial \tilde{\theta}_{k-1}^{w}} \qquad (3.19)$$

where η_1, η_2, η_3 is the learning rates of the semantic separation layer , the POS separation layer and sentiment polarity separation layer respectively. X_w is concatenated by $w_{t-n}, w_{t-n+1}\ldots, w_{t+2}, w_{t+n}$.The learning rate of parameter X_w is η_4,and its updating method shows in the following

$$X_w \leftarrow X_w + \eta_4 \frac{\partial L\left(\theta_{i-1}^{w}, \hat{\theta}_{j-1}^{w}, \tilde{\theta}_{k-1}^{w}, X_w\right)}{\partial X_w} \qquad (3.20)$$

From the above derivation process, we can get the following algorithm for updating the parameters of PSWV-model in hierarchical softmax.

```
Input: the word vectors $X_{w_{t-n}}, X_{w_{t-n}},\ldots, X_{w_{t+n-1}}, X_{w_{t+n}}$ of context of word $w_t$
1). $\theta_{i-1}^{w} = 0$, $\hat{\theta}_{j-1}^{w} = 0$, $\tilde{\theta}_{k-1}^{w} = 0$
2). $X_{w_t} = X_{w_{t-n}} \oplus X_{w_{t-n}} \oplus \ldots \oplus X_{w_{t+n-1}} \oplus X_{w_{t+n}}$($\oplus$ represents concatenating)
3).For i=2:$n^c$ ;j=2: $n^p$;k=2: $n^s$
{
$g1 = \eta 1[1 - \xi_i^w - \rho(X_w{}^T \theta_{i-1}^{w})]$;
$g2 = \eta 2[1 - \hat{\xi}_i^w - \rho(X_w{}^T \hat{\theta}_{j-1}^{w})]$;
$g3 = \eta 3[1 - \tilde{\xi}_i^w - \rho(X_w{}^T \tilde{\theta}_{k-1}^{w})]$;
$\theta_{i-1}^{w} \coloneqq \theta_{i-1}^{w} + g1X_w$;
$\hat{\theta}_{j-1}^{w} \coloneqq \hat{\theta}_{j-1}^{w} + g2X_w$;
$\tilde{\theta}_{k-1}^{w} \coloneqq \tilde{\theta}_{k-1}^{w} + g3X_w$;
$X_w := X_w + \eta 4(g1\theta_{i-1}^{w} + g2\hat{\theta}_{j-1}^{w} + g3\tilde{\theta}_{k-1}^{w})X_w$;
}
```

3.5 Time complexity

The word vector models in the literature include HLBL word vector (hierarchical log-bilinear word vector) [11], Turian's word vector, C & W word vector [12], the word vector of Huang and CBOW and Skip-gram. The structure of CBOW and Skip-gram is more simple and clear, so they are more efficient and can be used as benchmark for comparison. The time complexity is defined as the amount of parameters that need to be accessed during each iteration step. The time complexity of the CBOW can be defined as:

$$TC(CBOW) = 2nm + mlog_2|V| \tag{3.21}$$

m is the dimension of word vector. $|V|$ is the size of dictionary. n is the size of training window of word vector. 2nm is the amount of parameters needing to be accessed in input layer. $\mathrm{mlog_2|V|}$ is the amount of parameters needing to be accessed in Hierarchical softmax in output layer.

The time complexity of the PSWV-model can be defined as:

$$TC(PSWV) = 2nm + mnlog_2(1 + \lambda_2 + \lambda_3)|V| \tag{3.22}$$

where λ_2, λ_3 is the proportion of nodes increased after adding the POS separation layer and sentiment polarity separation on the basis of CBOW-model. Therefore, the ratio of time complexity of PSWV-model to time complexity of CBOW is:

$$TCR = \frac{2nm+mnlog_2(1+\lambda_2+\lambda_3)|V|}{2nm+mlog_2|V|} = \frac{2n+nlog_2(1+\lambda_2+\lambda_3)|V|}{2n+log_2|V|} \tag{3.23}$$

In theory, the minimum upper bound and maximum lower bound for increasing the number of nodes by adding POS separation layer are $sup\ (\lambda_2) = 5, inf\ (\lambda_2) = 0$ respectively. If we adopt the five grades separation structure shown in Fig3.2, the minimum upper bound and maximum lower bound for increasing the number of nodes by adding sentiment polarity separation layer are $sup\ (\lambda_2) = 4, inf\ (\lambda_2) = 0$ respectively. Hence, the minimum upper bound of TCR is

$$sup\ (TCR) = \frac{2n+nlog_2 10|V|}{2n+log_2|V|} \tag{3.24}$$

The maximum lower bound of TCR is

$$\mathrm{inf\ (TCR)} = \frac{\mathrm{2n+nlog_2 0|V|}}{\mathrm{2n+log_2|V|}} \tag{3.24}$$

It can be seen that the addition of POS separation layer and sentiment polarity separation layer on the basis of CBOW does not increase the time complexity unacceptably. For example, when the size of dictionary is $|V|$ =130000 and the size of training windows is $n = 5$, the value of $\mathrm{sup(TCR)}$ and $\mathrm{inf(TCR)}$ are

$$sup(TCR) = 4.13 \tag{3.25}$$

$$inf(TCR) = 3.58 \tag{3.26}$$

4. Experiment and evaluation

In the contemporary literature, the classical word vector models include CBOW, SKIP-GRAM, GLOVE, and HLBL (hierarchical log-bilinear) word vectors, Turian's vector, C&W word vector and Huang word vector. The common test method for the quality of word vectors is added to some natural language processing tasks as specific features, such as Named Entity Recognition and sentiment polarity analysis. We compare and analyze the performance of PSWV model and other models on the tasks of named entities recognition and sentiment polarity recognition.

4.1 Performance of PSWV-model in task of Named Entity Recognition

In this study, English corpus of Wikipedia (snapshot as of December 2017) and Reuters RCV data training set are used to train PSWV-model. Firstly, the data sets are cleaned and normalized, such as deleting short sentences, removing abnormal sentences, converting uppercase into lowercase. The Arabic numeral and the low frequency word is removed. Finally, the data set is integrated into the 15billion text dataset. Then the whole data set is annotated in part of speech using Stanford Pos-tagger. After that the text data is annotated according to sentiment lexicons such as LIWC(Pennebaker,2007),MPOA Subjectivity cues lexicon(Riloff and Wiebe 2003),Bing Liu opinion lexicons(Bing Liu,2004),Sentiword Net(Stefano,2010). In this experiment, the dimension size of the word vector is $m = 50$, and the size of the training window is $n = 3$.

Named Entity Recognition recognizes the boundaries and types of named entity in text. It plays a basic role in information extraction, question answering system, machine translation, and can be used to evaluate the quality of word vector models. Okazaki in 2007 proposed a fast implementation of conditional Random Fields to evaluate the performance of Named Entity Recognition system. In this study, we chose Okazaki's standard to evaluate the quality and performance of word vector models as follows:

$$F1 = \frac{2PR}{(R+P)} \tag{4.1}$$

where P is the Accuracy for block or named entity. R is recall rate for block or named entity. We are using the program of Okazaki (Ner.py) to obtain the commonly used features, and add the word vector on the basis of these common features. We are comparing the word vector such as CBOW, Skip-gram, Turia and Huang with our PSWV model. The experimental results are shown in Tab.4.1. The test data is based on the publicly evaluated task dataset (CoNLL 2003) from Reuters News [13].

Tab.4.1 Four categories Performance of F1 score based on CRFSuite by Okazaki

dataset &Task	Huang	Turian	HLBL	CBOW	CDNV	PSWV
data:TEST	83.18	86.14	85.62	86.45	87.12	92.16*
Task1:LOC	87.76	88.90	90.62	89.04	90.60	96.81*
Task2:MISC	77.24	77.72	78.33	78.29	78.60	88.84*
Tak3:ORG	83.50	83.11	83.61	83.26	83.28	93.64*
Task4:PER	93.92	93.84	94.24	94.21	94.38	98.87*

The results show that all models can improve the performance of named entity recognition system based on CRF. Tab.4.1 reflects that PSWV-model can significantly improve the performance of named entity recognition system based on CRF when adding POS separation layer and sentiment polarity separation layer. The performance on the test data set is significantly superior to the benchmark systems CBOW. The performance of PSWV-model on the sub-tasks including LOC, MISC, ORG and PER is significantly better than that of the benchmark system CBOW word vector and other word vectors. CDNV and PSWV have adopted the frame of part of speech separation at the same time. CDNV adopted the Dissociation between Nouns and Verbs. PSWV adopted a completely separate structure of part of speech. Hence, the advantage of PSWV is not larger than CDNV. However, the advantage of PSWV is relatively larger than that of other models which do not adopt the separation of part of speech.

4.2 Performance of PSWV-model in task of sentiment polarity analysis

For an additional mode of verification of, the performance of PSWV model in sentiment analysis, we use labeled micro-blog data to verify the effect of PSWV-model on task of sentiment polarity recognition when applying deep learning (LSTM) model. The hot topics in our study are retrieved from Sina Weibo (Fackbook and Twitter). According to the retrieval results, we take the reviews from popular and authority newspapers or website such as Beijing News, People's Daily and other news media. We collected more than 500 thousand reviews or comments from these websites through data collecting software—octopus12.0. And then we were able to obtain more than 300 thousand reviews or comments after data cleaning process. The data cleaning process included removing the user names, empty comments, and less than 5 character comments. The user names and empty comments contain no sentiment information, and most of the comments with less than five characters do not contain emotional information. This data will become noise data and interfere with the results of experiments when doing emotional analysis. So the three kinds of data were deleted to ensure the accuracy of the experiment. After that, the reviews text data were labeled with emotion signs. And the data was divided into 4 categories and 21 subclasses according to emotion labeling. 10000 reviews with positive sentiment polarity and 10000 with negative sentiment polarity were randomly selected as training sets after sentiment tagging of the text data. And then 2500 items were randomly selected as test sets in the remaining data. Hence, there are 20000 reviews in training data and 5000 reviews in test data set.

In our study, the performance of the model is measured by three indexes: accuracy rate, precision prediction, recall rate and F1 value. Mixture matrix is introduced to define the three indexes. Accuracy rate is the rate of correctly predicted reviews number accounts for the total number of test data sets as following

$$Accuracy\ rate = \frac{TP+TN}{TP+FP+TN+FN} \tag{4.2}$$

where TP (True Positive) is the number of positive comments that are correctly predicted as positive ones. FP (False Positive) is the number of negative emotion comments that are predicted as negative ones. FN (False Negative) is the number of positive emotion comments that are predicted as negative ones. TN (True Negative) is the number emotion of positive comments that are predicted as negative ones.

Take positive emotion comments as an example to explain the meaning of Precision, Recall and F1 values. In the test data set, Precision is the rate of the number of reviews correctly predicted as positive emotions and the total number of reviews predicted as positive emotions:

$$Precision = \frac{TP}{TP+FP} \tag{4.3}$$

Recall is the rate of the number of reviews correctly predicted as positive emotions and the total number of positive emotions reviews

$$\text{Recall rate} = \frac{\text{TP}}{\text{TP+FN}} \tag{4.4}$$

F1 value is defined as the harmonic average of precision and recall as following

$$F1 = \frac{2\times Recall\times Precision}{Recall+Precision} \tag{4.5}$$

We input CBOW, Glove, CDNV and PSWV word vectors as features into the deep learning model (LSTM [14]) respectively to predict the sentiment polarity of comments or reviews. Tab.4.2 shows the experimental results when dimensions of word vectors are 50,100,150,200,250,300. The following steps are adopted in the experiment.

1) Setting the initial value of the super parameters in the LSTM neural network. The initial

iteration is set as 3. The loss function is set as the cross entropy loss function. The optimizer selects the Adma optimizer.

2) Training the LSTM neural network to predict the sentiment polarity with different word vector of comments which are obtained by CBOW, Glove, CDNV and PSWV models.

3) Carrying out several experiments of LSTM under each word vector model in different dimensions such as 50,100,150,200,250,300. And the super parameters in the model are constantly adjusted to improve the performance of LSTM model. After many experiments, the performance evaluation indexes of different word vectors in LSTM neural network are obtained, as shown in Tab.4.2.

Tab.4.2 The accuracy rate of sentiment polarity recognition based on LSTM Neural Network Model under different word Vector models

dimension Word vector	50	100	150	200	250	300
CBOW	81.1	81.9	82.3	83.3	*83.8	82.8
CDNV	83.1	84.3	85.1	*85.3	84.2	83.8
GLOVE	81	83.70	83.6	*84.8	82.7	83
PSWV	85.1	86.3	91.6	92.5	87.1	85

According to Tab.4.2 and Fig.4.1, we can see that the performance of sentiment polarity recognition by LSTM model using PSWV word vector in is better than that using CBOW, Glove and CDNV in different dimensions. The reason being that the PSWV word vector not only uses the semantic information provided by the context and the words order information and part of speech information, but also makes use of the sentiment information of the words during the training process. Therefore, the performance of sentiment polarity recognition in different dimensions is relatively good than other models.

Comparing with CBOW in the process of word vector training, CDNV not only uses the semantic information provided by context, but also uses the separation information of nouns and verbs. Therefore, it is better to capture the sentiment polarity of words compared with CBOW. In addition, CBOW use the several words of the context to find the probability of the specified word, so CBOW can just capture the local information of the context. However, Glove model trains the word vector with global information, so the Glove word vector is better than CBOW word vector.

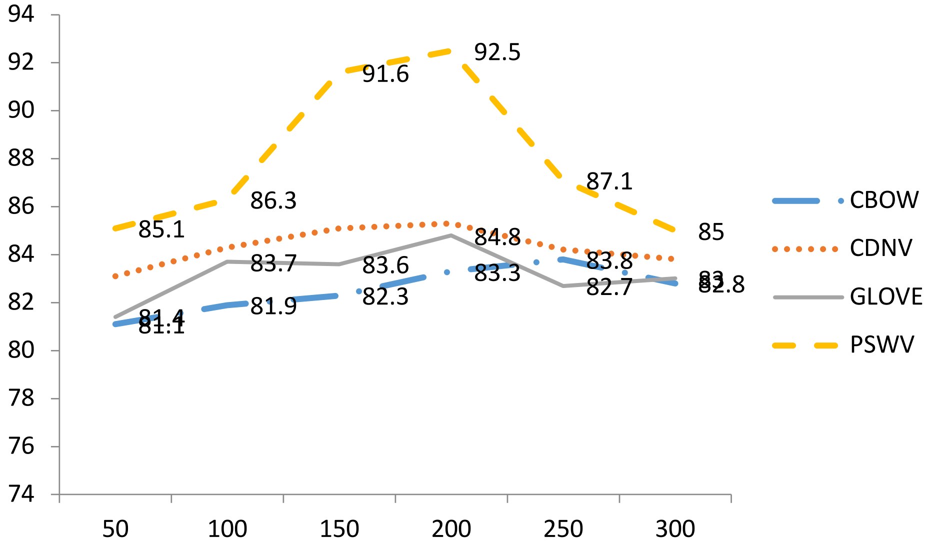

Fig.4.1 The accuracy rate of sentiment polarity recognition based on LSTM Model under different word Vector models and different dimensions

5. Conclusions and Forecast

Language models play important role in practical applications of NLP task such as machine translation, information indexing, voice recognition, context processing. Hence, we analyze the principles of traditional statistical language models, such as N-gram model, N-POS model and the maximum entropy model. We find that statistical language model has some inherent weaknesses, such as semantic gap and Features sparse phenomenon. Semantic gap phenomenon shows that the strings of two semantically similar words may be completely different. But there are also some advantages, such as that the model can get many kinds of features of language including part of speech features, word order features, word combination features and so on.

Contrary to that, the representation of words in neural Network Language Model is vector form which is semantic oriented. And the distance of two semantic similar words vector are close, which can solve the problem of semantic gap in traditional statistical language model. The development trend of neural network model is more and clearer after Bengion's work in 2002. In 2013, two improved neural network language models CBOW and Skip-Gramm proposed by Mikolov push the application of neural network model language apply rapidly. Mikolov improves Bengion neural network model to reduce the computational complexity of the model greatly. But the improvement of changing concatenating vectors in the hidden layer into calculating the sum of vectors in the projection layer results in the loss of the original word order features. In addition, CBOW and Skip-Gram cannot capture the linguistic features and emotional information. However, the improvement of Mikolov provides a good idea for other researchers. For example, Hierarchical softmax algorithm can significantly reduce the computational complexity. In the framework of Mikolov's model, the CDNV-model proposed by the scholars Hu in 2016 attempts to utilize the part-of-speech features of verbs and nouns, and achieves good results. Inspired by the traditional statistical language model and neural network model, our attempt is to put forward PSWV-model in order to use more language information such as word order features, part of speech features and sentiment polarity information under the framework of Mikolov model. And then we can compare and analyze some advantages of PSWV model and other models including CBOW and Skip-Gram, CDNV in the NLP tasks including named entities recognition and sentiment polarity recognition.

In near future, further development may be made in the following aspects. First, more language features can be accommodated under the framework of CBOW. And more deep learning models such as RNN, CNN, LSTM+CNN etc. can be used to verify the performance of word vector model in some NLP tasks such as sentiment polarity recognition. Second, we can try to further develop the model on the Skip-Gram architecture. Third, the computational complexity of the model may be reduced further.

第 3 章 基于深度学习 LSTM-CNN 模型的旅游评论短文本情感分析

3.1 基于浅层学习的旅游评论短文本情感分析模型

在传统的文本情感分析领域中，有大量的研究论文采用的是基于浅层学习算法比如逻辑回归模型（Logistic Regression)、支持向量机模型（SVM，Support Vector Machine）和朴素贝叶斯模型（Naive Bayesian）等。传统的文本情感分析主要的做法是先对数据进行人工分析，采用人工的方式去提取文本数据中的特征，然后将这些浓缩的特征输入到以上所提到的各类浅层学习算法模型中去进一步抽取情感倾向。然而在当前海量大数据时代，采用人工方法去抽取每时每刻新增的海量文本的特征已经是不可能完成的任务。本章的目的是使用浅层学习算法对语言模型输出的词向量构造的数据进行情感分类，结果将与深度学习算法产生的结果作对比。为了引入本章的主要内容基于深度学习的情感分析模型，我们有必要对传统的文本情感分析的主要模型包括逻辑回归模型（Logistic Regression)、支持向量机模型（SVM，Support Vector Machine）和朴素贝叶斯模型（Naive Bayesian）包括做一个回顾分析。

3.1.1 基于支持向量机的旅游评论短文本情感分析模型

支持向量机（SVM，Support Vector Machine）是一种有监督学习算法由 Cortes and Vapnik 在 1995 年正式提出[1]，1998 年 Joachimes 发现 SVM 在文本分类中拥有非常卓越的表现[2]，后来 SVM 就被大量的学者广泛地应用在文本情感、文本分类的各类任务中[38-40]。支持向量机属于线性分类器，这类分类器能够使经验误差最小[41]。本次研究使用 SVM 做情感分类任务，利用 SVM 分类模型对词向量提供的数据特征进行情感分类，具体做法是根据提供的训练集数据将模型参数学习到最优，然后利用训练好的模型对测试集数据中正向情感和负向情感的评论作出预测，最后根据预测结果判断训练模型的好坏。支持向量机泛化能力强，能够很好地处理高维数据，无局部极小值问题，在数据集较小的分类任务中效果较好，不足之处是难于处理大数据量的分类任务，对缺失数据敏感，而且要寻找一个合适的核函数很困难。

3.1.2 基于罗杰斯特回归的旅游评论短文本情感分析模型

罗杰斯特回归（Logistic Regression）模型是一个二分类模型[44]，与本次研究将正负情感数据进行分类的任务相符。罗杰斯特回归本质上是线性回归模型，只是在特征进行线性求和之后，利用 Sigmoid 函数将结果映射到 0 到 1 之间。罗杰斯特回归实现简单，计算复杂度较低，计算过程中占用的计算资源较少，不足之处是容易造成欠拟合，准确率不是很高，当特征空间较大时分类效果不是很好。

3.1.3 基于朴素贝叶斯的旅游评论短文本情感分析模型

3.2 基于深度学习的情感分析

3.2.1

本章引入两个深度学习模型，长短时记忆网络（Long Short Term Memory Network，LSTM）和卷积神经网络（Convolutional Neural Network，CNN）。自 2006 年开始流行以来，首先在图像处理和语音识别领域中取得了不错的成绩，随着研究的深入，越来越多的自然语言处理任务也开始使用深度学习模型解决，而在深度学习模型中最长使用的是 LSTM 和 CNN。

[1] C. Cortes, VN Vapnik. Support Vector Networks[J]. Machine Learning, 1995, 20(3):273-297.

[2] Joachims T . Text Categorization with Support Vector Machines: Learning with Many Relevant Features[C].Proc Conference on Machine Learning. Springer, Berlin, Heidelberg, 1998.