

User Rating Classification via DBN and Sentiment Analysis

Princy Baby¹, Krishnapriya B²

P.G. Student, Department of Computer Science and Engineering, Government Engineering College, Idukki, Kerala, India¹

Associate Professor, Department of Computer Science and Engineering, Government Engineering College, Idukki, Kerala, India²

ABSTRACT: User-generated texts such as reviews are valuable sources of emotional information. Therefore, the rating of a review is one of the reliable factors for all users to read and trust the reviews. User comments are important for recommender systems because they include various types of emotional information that may influence the correctness or precision of the recommendation. A deep learning model is used to process user comments and to generate a possible user rating for user recommendations. The system uses sentiment analysis to create a feature vector as the input nodes. Next, the system implements noise reduction in the data set to improve the classification of user ratings. A deep belief network and sentiment analysis (DBNSA) achieves data learning for the recommendations. The user rating prediction is obtained from the output layer of the DBN. The timeline is also an important factor since users are affected by their previous experience with similar products or services. The timeline is also one of the factors used in dealing with the cold-start problem user.

KEYWORDS: Classification, deep learning, Sentiment Analysis.

I. INTRODUCTION

Machine learning has received greater attention with the success of deep learning. Deep learning can create deep models of complex multivariate structures in structured data. Deep in many fields, with significant successes in many applications. Convolution neural networks[1], deep belief networks (DBNs)[2], and many other approaches have been proposed to enhance the abilities of deep structure networks.

The emotional context in recommender systems[6], arguing that emotions are crucial for users' decision making in the recommendation process. Users usually transmit their decisions together with emotions. SenseNet using a linguistic tool to identify the polarity values of words. In this approach, using WordNet, they manually assign a numerical value to some verbs, adjectives, and adverbs. They then detect their polarity values using ConceptNet.

DBNSA method is using user comments to classify user ratings. User ratings are determined by the numerical value in which the users give to a product or service. Generally, there are two types of classifiers, discriminative and generative. Discriminative classifiers build a function from an input set to a class label. Generative classifiers make a model of a joint probability and predict the class label of an input instance. The DBN and sentiment analysis (DBNSA) method is a discriminative classifier, which predicts the rating probability from a word vector generated by WordNet[10] sentiment analysis and then uses deep learning to train a model to predict the rating. Many methods for classifying user ratings have been proposed, including libSVM and multilayer perceptron (MLP). DBNSA model to enhance deep learning classifier capability through preprocessing and compare our results to baseline methods such as the libSVM and MLP algorithms.

The data set consists of a set of users, $U = u_1, u_2, \dots, u_m$ and a set of businesses, $B_u = bu_1, bu_2, \dots, bu_n$, where m is the number of users, and n is the number of businesses. A rating for each business or product bu_j by user u_i is expressed by ratings $R = R[U, B_u]_{m \times n}$. This matrix R denotes the rating of user u_i on business or product bu_j where i is an index of the user and j is an index of businesses. The elements of R could be any number in the range of 15, as it was collected from the database.

The timeline is also an important factor since users are affected by their previous experience with similar products or services. The timeline is also one of the factors used in dealing with the cold-start problem user. Another timeline could contain information to help solve the cold-start problem.

II. RELATED WORK

Feature Selection Methods

Feature Selection methods can be divided into lexicon-based methods that need human annotation, and statistical methods which are automatic methods that are more frequently used. Lexicon-based approaches usually begin with a small set of seed words. Then they bootstrap this set through synonym detection or online resources to obtain a larger lexicon. The feature selection techniques treat the documents either as a group of words (Bag of Words (BOWs))[5] or as a string that retains the sequence of words in the document. BOW is used more often because of its simplicity for the classification process. The most common feature selection step is the removal of stop-words and stemming (returning the word to its stem or root i.e. flies fi fly)

Sentiment Classification Techniques

Sentiment Analysis learns the positivity or negativity of products or services from user comments. Sentiment Analysis is based on an opinion lexicon. An opinion lexicon is a dictionary[11] containing words, which express the polarity of words through positive or negative Sentiments, such as happy, good, bad, or disgusting. These opinion words are used in Sentiment Analysis as the key indicator to calculate the opinion of the user.

Sentiment Analysis learns the positivity or negativity of products or services from user comments. Sentiment Analysis is based on an opinion lexicon. An opinion lexicon is a dictionary containing words, which express the polarity of words through positive or negative Sentiments, such as happy, good, bad, or disgusting. These opinion words are used in Sentiment Analysis as the key indicator to calculate the opinion of the user. Y. Choi and C. Cardie [1] Introduced heuristic-based methods for determining the polarity of a Sentiment bearing expression. Each assesses the polarity of the words or constituents using a polarity lexicon that indicates whether a word has positive or negative polarity, and finds negators in the given expression using a negator lexicon. The methods then infer the expression-level polarity using voting-based heuristics or heuristics that incorporate compositional semantics. The lexicons are described Compose first checks whether the first argument is a negator, and if so, flips the polarity of the second argument. Otherwise, Compose resolves the polarities of its two arguments. Note that if the second argument is a negator, do not flip the polarity of the first argument, because the first argument, in general, is not in the semantic scope of the negation. Instead, the second argument as a constituent with negative polarity. Experiment with two variations of the Compose function depending on how conflicting polarities are resolved: COMPOMC uses a Compose function that defaults to the Majority Class of the polarity of the data, while COMPOPR uses a Compose function that selects the polarity of the argument that has higher semantic Priority. For brevity, we refer to COMPOPR and COM-POMC collectively as COMPO.

Phrase Dependency Parsing

Y. Wu, Q. Zhang, X. Huang, and L [2] Introduced a Dependency grammar is a kind of syntactic theories presented. In dependency grammar, the structure is determined by the relation between a head and its dependents. In general, the dependent is a modifier or complement; the head plays a more important role in determining the behaviors of the pair. Therefore, criteria of how to establish dependency relations and how to distinguish the head and dependent in such relations are a central problem for dependency grammar. There are seven pairs of dependency relationships, depicted by seven arcs from heads to dependents. The mainstream of dependency parsing is conducted on lexical elements: relations are built between single words. A major information loss of this word-level dependency tree compared with the constituent tree is that it doesn't explicitly provide local structures and syntactic categories of phrases. On the other hand, the dependency tree provides connections between distant words, which are useful in extracting long distance relations.

III. METHODOLOGY

The system begins by extracting user comments from the data set. The system applies word stemming to stem all the words from the sentences in the user comments. We used a sentiment analysis word dictionary of positive and negative opinion words and the bag of words algorithm to count the words. The 50-D input vector is created from the WordNet sentiment analysis.

Pre-processing refers to the transformations applied to our data before feeding it to the algorithm. Data Pre-processing is a technique that is used to convert the raw data into a clean data set. In other words, whenever the data is gathered from different sources it is collected in raw format which is not feasible for the analysis. For achieving better results from the applied model in Machine Learning projects the format of the data has to be in a proper manner. Some specified Machine Learning model needs information in a specified format, for example, Random Forest algorithm does not support null values; therefore to execute random forest algorithm null values have to be managed from the original raw data set. Another aspect is that the data set should be formatted in such a way that more than one Machine Learning and Deep Learning algorithms[22] are executed in one data set, and the best out of them is chosen.

GloVe[13], coined from Global Vectors, is a model for distributed word representation. The model is an unsupervised learning algorithm for obtaining vector representations for words. This is achieved by mapping words into a meaningful space where the distance between words is related to semantic similarity. Training is performed on aggregated global word-word co-occurrence statistics from a corpus, and the resulting representations showcase interesting linear substructures of the word vector space. It is developed as an open-source project at Stanford. As the log-bilinear regression model for unsupervised learning of word representations, it combines the features of two model families, namely the global matrix factorization and local context window methods.

The gloVe can be used to find relations between words like synonyms, company - product relations, zip codes, and cities, etc. It is also used by the spacy model to build semantic word embeddings/feature vectors while computing the top list words that match with distance measures such as Cosine Similarity and Euclidean distance approach. It was also used as the word representation framework for the online and offline systems designed to detect psychological distress in patient interviews. The open-source knowledge graph Weaviate uses a GloVe based vectorization model to store data objects. It requires that the input data be integer encoded so that each word is represented by a unique integer. This data preparation step can be performed using the Tokenizer API also provided with Keras. The Embedding layer is initialized with random weights and will learn an embedding[14] for all of the words in the training dataset. The Embedding layer is defined as the first hidden layer of a network.

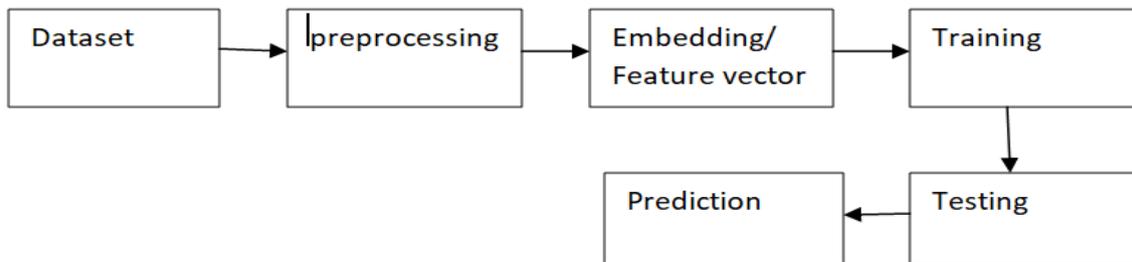


Fig 1: An architecture of the overall system

Deep Belief Networks consist of multiple layers with values, wherein there is a relation between the layers but not the values. The main aim is to help the system classify the data into different categories. The First Generation Neural Networks used Perceptrons which identified a particular object or anything else by taking into consideration weight or pre-fed properties. However, the Perceptrons[21] could only be effective at a basic level and not useful for advanced technology. To solve these issues, the Second Generation of Neural Networks saw the introduction of the concept of Backpropagation in which the received output is compared with the desired output, and the error value is reduced to zero. Support Vector Machines created and understood more test cases by referring to previously input test cases. Next came directed cyclic graphs called belief networks which helped in solving problems related to inference and learning problems. This was followed by Deep Belief Networks[16] which helped to create unbiased values to be stored in leaf nodes. The first step is to train a layer of properties which can obtain the input signals from the pixels directly. The next step is to treat the values of this layer as pixels and learn the features of the previously obtained features in a second

hidden layer. Every time another layer of properties or features is added to the belief network, there will be an improvement in the lower bound on the log probability of the training data set.

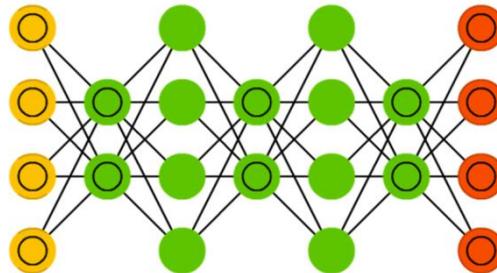


Fig 2: Deep Belief Networks.

The Test Dataset is only used once a model is completely trained (using the train and validation sets). The test set is generally what is used to evaluate competing models. It includes only input data, not the corresponding expected output. The testing data is used to assess how well your algorithm was trained, and to estimate model properties. The test dataset is a dataset used to provide an unbiased evaluation of a final model fit on the training dataset.

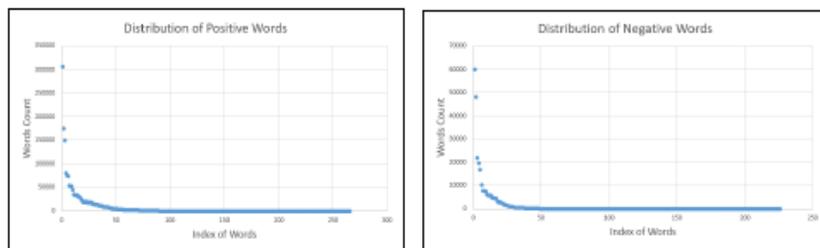
Prediction is the last step where the input is the text file that the user is uploaded. The input text file is pre-processed, trained, and classified to obtain the intent/domain. Based on the intent/domain identified, the sentences are loaded from the trained dataset. Accuracy is defined as the percentage of correct predictions for the test data. It can be calculated easily by dividing the number of correct predictions by the number of total predictions.

IV. EXPERIMENTAL RESULTS

Input Feature Vector

In this section, describe the distribution of positive and negative words that appear in user comments. Extracted all opinion words from user comments and used the GloVe algorithm to count the opinion words. Fig. 4 shows the distribution of negative words. Figs. 6 and 7 are created from the statistical bag of words, for which the index of words is opinion words from the sentiment analysis dictionary. There are 266 positive and 226 negative opinion words. For example, positive opinion words are "good," "great," and "nice." Negative opinion words are "never," "bad," and "old." As described in our proposed methodology section, we use a 50 dimension feature vector as our input node. Fig. 3 shows that most positive opinion words fall within the first 50 words, while Fig. 4 shows that there are less than 50 negative opinion words. To make the input feature vector less sparse, we consider the minimum value which that is significant for use as Fig. 4. Distribution of negative words. input vector. Take 25 dimensions from positive and negative opinion words as our input vector.

Figures show the results of Distribution of positive and negative words. (a) shows the distribution of positive words (b) is the image of Distribution of negative words. Take 25 dimensions from positive and negative opinion words as our input vector.



(a)

(b)



Experimental Results with Data Set

Table 1 gives the comparison of the test data in the test data set, which comprises 30% of the data. Test the trained model across all methods with the test data. The prediction of user rating is measured by accuracy. Find that the DBNSA has the best value among the baseline methods for the YELP data set. The YELP dataset distribution has more good impression user ratings. This means that DBNSA is very good at predicting the convergence distribution for data, which has more convergence features into one of its classes. In the Amazon data sets, which are more similar to each other than to YELP, the DBNSA still yields better results than the other baseline methods but does not perform as well as with the YELP data set. These two data sets have more neutral impressions (three-star ratings) than the YELP data set. In the Amazon data set, DBNSA has the best accuracy value, 75.85%, followed by CNN with 65.87%.

Table 1 gives a comparison of Results between methods. The prediction of user rating is measured by accuracy. The DBNSA has the best value among the baseline methods for the different data set.

Dataset	Method	Accuracy
Yelp	MLP	61.31
	libSVM	62.58
	CNN	63.81
	DBNSA	72.83
Amazon	MLP	75.85
	libSVM	55.63
	CNN	65.87
	DBNSA	75.85

Table 1 : Comparison of results between methods.

Table 2 gives a comparison of Results between datasets. The prediction of user rating is measured by accuracy in Epochs. The Timeline comments used Dataset has the best accuracy among the DBNSA method for the different data sets

Dataset	Method	Epochs	Accuracy
Yelp	DBNSA	11	65.31
Amazone			72
Timeline Comments			77

Table 2: Comparison of accuracy results between datasets.

VI. CONCLUSION

A deep learning model is used to process user comments and to generate a possible user rating for user recommendations. The system uses sentiment analysis to create a feature vector as the input nodes. Next, the system implements noise reduction in the data set to improve the classification of user ratings. A deep belief network and sentiment analysis (DBNSA) achieves data learning for the recommendations. The user rating prediction is obtained from the output layer of the DBN. The timeline is also an important factor since users are affected by their previous experience with similar products or services. The timeline is also one of the factors used in dealing with the cold-start problem user.

REFERENCES

[1] Y. Choi and C. Cardie, Learning with compositional semantics as structural inference for subsentential Sentiment Analysis, in Proc. Conf. Empirical Methods Natural Lang. Process., Stroudsburg, PA, SENTIMENT ANALYSIS, Oct. 2008, pp.793801.

- [2] Y. Wu, Q. Zhang, X. Huang, and L. Wu, Phrase dependency parsing for opinion mining, in Proc. Conf. Empirical Methods Natural Lang.Process. (EMNLP), vol. 3, Aug. 2009, pp. 1533-1541.
- [3] N. Jakob and I. Gurevych, Extracting opinion targets in a single-and cross-domain setting with conditional random fields, in Proc. Conf. Empirical Methods Natural Lang. Process. (EMNLP), Oct. 2010, pp. 1035-1045.
- [4] W. Medhat, A. HasSentiment Analysis, and H. Korashy, Sentiment Analysis algorithms, and applications: A survey, Ain Shams Eng. J., vol. 5, no. 4, pp.1093-1113, Dec. 2014.
- [5] T. Mikolov, I. Sutskever, K. Chen, G. Corrado, and J. Dean, Distributed representations of words and phrases and their compositionality, in Proc. Adv.Neural Inf. Process. Syst., Oct. 2013, pp. 3111-3119.
- [6] G. Zhao, X. Qian, and X. Xie, User-service rating prediction by exploring social users' rating behaviors, rating behaviors, IEEE Trans.Multimedia, vol.18, no. 3, pp. 496-506, Mar. 2016.
- [7] Z. Hai, G. Cong, K. Chang, P. Cheng, and C. Miao, Analyzing Sentiments in one go: A supervised joint topic modeling approach, IEEE Trans. Knowl.Data Eng., vol. 29, no. 6, pp. 1172-1185, Jun. 2017.
- [8] Y. Fang, H. Tan, and J. Zhang, Multi-strategy Sentiment Analysis of consumer reviews based on semantic fuzziness, IEEE Access, vol. 6, p. 2062520631, 2018.
- [9] Y. Ma, Z. Xiang, Q. Du, and Fan, Effects of user-provided photos on hotel review helpfulness: An analytical approach with deep learning, Int. J. Hospitality Manage., vol. 71, pp. 120-131, Aug. 2018.
- [10] B. Purkaystha, T. Datta, M. S. Islam, and M.-E-Jannat, Rating prediction for recommendation: Constructing user-profiles and item characteristics using backpropagation, Appl. Soft Comput., vol. 75, pp. 310-322, Feb. 2019
- [11] Rung-Ching Chen and Hendry User Rating Classification via Deep Belief Network Learning and Sentiment Analysis IEEE TranSentimentAnalysisction On Computational social System, 2019.
- [12] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
- [13] K. F. Yeung, Y. Yang, and D. Ndzi, "Context-aware news recommender in mobile hybrid p2p network," in Proc. 2nd Int. Conf. Comput. Intell. Commun. Syst. Netw., Jul. 2010, pp. 54–59.
- [14] G. Qiu, B. Liu, J. Bu, and C. Chen, "Opinion word expansion and target extraction through double propagation," *Comput. Linguistics*, vol. 37, no. 1, pp. 9–27, Mar. 2011. [Online]. Available: <http://aclweb.org/anthology/J11-1002>
- [15] Y. Wu, Q. Zhang, X. Huang, and L. Wu, "Phrase dependency parsing for opinion mining," in *Proc. Conf. Empirical Methods Natural Lang.Process. (EMNLP)*, vol. 3, Aug. 2009, pp. 1533–1541. [Online]. Available: <http://dl.acm.org/citation.cfm?id=1699648.1699700>.
- [16] R. Agrawal, T. Imieliński, and A. Swami, "Mining association rules between sets of items in large databases," *SIGMOD Rec.*, vol. 22, no. 2, pp. 207–216, Jun. 1993. doi: 10.1145/170036.170072
- [17] P. Resnick, N. Iacovou, M. Suchak, P. Bergstrom, and J. Riedl, "GroupLens: An open architecture for collaborative filtering of netnews," in Proc. ACM Conf. Comput. Supported Cooperative Work (CSCW), Oct. 1994, pp. 175–186. doi. 10.1145/192844.192905.
- [18] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Item-based collaborative filtering recommendation algorithms," in *Proc. 10th Int. Conf. World Wide Web*, May 2001, pp. 285–295. doi: 10.1145/371920.372071.
- [19] S. Jiang, X. Qian, J. Shen, Y. Fu, and T. Mei, "Author topic model-based collaborative filtering for personalized POI recommendations," *IEEE Trans. Multimedia*, vol. 17, no. 6, pp. 907–918, Jun. 2015.
- [20] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl, "Evaluating collaborative filtering recommender systems," *ACM Trans. Inf. Syst.*, vol. 22, no. 1, pp. 5–53, Jan. 2004. doi: 10.1145/963770.963772.
- [21] D. Ames, L. B. Maissen, and J. Brockner, "The role of listening in interpersonal influence," *J. Res. Pers.*, vol. 46, no. 3, pp. 345–349, Jun. 2012. [Online]. Available: <http://www.sciencedirect.com/science/article/pii/S0092656612000116>.
- [22] G. Zhao, X. Qian, and X. Xie, "User-service rating prediction by exploring social users' rating behaviors, "rating behaviors," *IEEE Trans.Multimedia*, vol. 18, no. 3, pp. 496–506, Mar. 2016.
- [23] X. Yang, H. Steck, and Y. Liu, "Circle-based recommendation in Online social networks," in *Proc. 18th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining (KDD)*, Aug. 2012, pp. 1267–1275. doi:10.1145/2339530.2339728.
- [24] M.-Y. Hsieh, W.-K. Chou, and K.-C. Li, "Building a mobile movie recommendation service by user rating and APP usage with linked data on Hadoop," *Multimedia Tools Appl.*, vol. 76, no. 3, pp. 3383–3401, Feb. 2017. doi: 10.1007/s11042-016-3833-0.
- [25] R. Salakhutdinov and A. Mnih, "Probabilistic matrix factorization," in *Proc. 20th Int. Conf. Neural Inf. Process. Syst. (NIPS)*, Dec. 2007, pp. 1257–1264. [Online]. Available: <http://dl.acm.org/citation.cfm?id=2981562.2981720>.