Proceedings of Seminar and Project

TITLE

SEMESTER

Oliver Wasenmüller and Prof. Didier Stricker
Department Augmented Vision
University of Kaiserslautern and DFKI GmbH

Introduction

The seminar and project TITLE (INF-XX-XX-S-X, INF-XX-XX-L-X) are continuative courses based on and applying the knowledge taught in the lectures 3D Computer Vision (INF-73-51-V-7) and Computer Vision: Object and People Tracking (INF-73-52-V-7). The goal of the project is to research, design, implement and evaluate algorithms and methods for tackling computer vision problems. The seminar is more theoretical. Its educational objective is to train the ability to become acquainted with a specific research topic, review scientific articles and give a comprehensive presentation supported by media.

In the XXX semester XXX, XXX projects addressing XXX were developed. Moreover, XXX seminar works addressed XXX. The results are documented in these proceedings.

Organisers and supervisors

The courses are organised by the Department Augmented Vision (`http://ags.cs.uni-kl.de`), more specifically by:

Oliver Wasenmüller
Prof. Dr. Didier Stricker

In the XXX semester XXX, the projects were supervised by the following department members:

NAME

MONTH YEAR

A sequence learning approach to sentiment analysis on call-center conversation data

Mohammad Baniasad[1] and Mohammad reza yousefi[2]

[1] m_baniasad14@cs.uni-kl.de
[2] yousefi@dfki.uni-kl.de

Abstract. Sentiment analysis is aiming to classify documents which are given in a single context, as expressing Positive/Negative sentiment towards a subjective topic. In general, sentiment analysis has been done on textual data and this field is mature and robust, with a lot of different approaches tried on it and surveys on these approaches[1]. Considering the context of call-center and the available voice data, in this paper we have tried to provide a sentiment analyzer on voice data. We have used a speech-to-text system to convert call data into text format and use conventional text classification methods on top of this data. Latest methods for sentiment classification has shown very good results, methods like ANN (artificial neural networks) and its variants like LSTM (Long Short Term Memory), Dynamic LSTM, Bidirectional LSTM and also statistical methods like Naive-Bayes[2] has been compared and analyzed in this paper as well as use of state of the art vector representation of words and documents like Word2vec[3] and Fast-text[4].
To analyze the potential lying in each of aforementioned methods a pipeline has been created. Three dataset that capture main characteristics that assumed to exists in transcribed call data has been selected and methods have been compared to each other using these data. Finally the call data has been fed to the pipeline and the results has been analyzed to show that due to nature of data and distribution of features and planned sentiment classes, the ANN based methods are superior to pure analytical based approaches and the best methods are those built on top of the new embedding algorithms like Word2Vec. For the purpose of getting the best results a preprocessing method (stemming) has also been tested with all the data and results shows stemming results in better performance in ANN based methods but reduced performance in pure analytical method.

Keywords: NLP, sentiment analysis, text classification

1 Introduction

Collecting information and feedback about product/service of a company has always been a vital necessity in the process of management and marketing. In today's industry using new automated and AI based approaches for analyzing the available raw data is the the key to having the lead in the market. One of the sources of these data that has not attracted a lot of attention yet, is calls that customers/clients are making/receiving to/from a company's call center. The blooming of activities in text classification, opinion mining and data mining in text, and maturity of speech-to-text systems, now lays the foundation of possibility to start the mining and classification of voice and call data.

T his paper is focusing on finding the correct way to approach this problem. Demonstrating it's challenges and possibilities. Also a survey on application of currently best known text sentiment classification methods on the automatic transcribed data from call data. Most of the the analyze on the methods are composed on a pipeline of (i)preprocessing (ii)feature selection (iii)sentiment classification.

In this paper we compare the currently most popular sentiment classification methods and a brief analysis of their performance on several datasets. In comparison with sentiment analysis papers the main contribution of this paper are (i)Comparison of sentiment analysis methods. (ii)Comparison of application of these methods on different languages. (German and English) (iii)Comparison of application of these methods on acquired real data from a call center of a German company. This paper is organized as following:
First section after introduction we talk about the chosen datasets, in order to create basic understanding about the analysis and results. Section 2 talks about experiments and it has been divided into 3 subsections.first experiment is done with an analytical method called Naive Bayes[2] to lay the foundations. 3-2 is about using ANN based methods and a choosing hyper parameters and analyzing the results. In the next subsection, 3-3 we 2 of explain the latest word and document representation techniques and their results in our dataset. At the end there would be the conclusion and suggestions for father works and the latest section would be acknowledgements

2 DATASETS

To be able to choose the best semantic analyzer methods we have used several sample datasets to try with chosen algorithms. Each dataset has it's own characteristics and the results should be analyzed with consideration of these characteristics. "IMDB movie reviews dataset[5]" is the most known dataset for comparison of different sentiment analysis algorithms which is a dataset of highly polar movie reviews (very likely to be against or in favor of subject). Movie reviews are in English and in average 1273 words per document. Next two dataset have been chosen from the product reviews of on-line retailer Amazon. The main difference between these two datasets are the language where one is in English and one is in German. The last dataset is the dataset gathered from automated transcribed German calls in the call center of a German company named Matelso GMBH. Here the sentiment positivity and negativity of a document has been mapped to the fact that the called client has been pursued to join a webinar or to meet the company's stand in a exposition or not. Labeling data for this dataset has been done by the callers of the company by declaring it success or fail with the implemented systems in the company using the phone keypad. This data is highly skewed toward negative data and small in sample size and for comparison special methods of application of system was incorporated. Table 1 shows the datasets and some of their features.

dataset	labeled items	Pos/Neg	unlabeled	description
English aclImdb reviews	50,000	train 12500/12500 test 12500/12500	50,000	Stanford University[5] highly polar movie reviews on IMDB
German amazon reviews	12,000	train 3000/3000 test 3000/3000	317,378	Weimar university [6] product reviews on Amazon
English amazon reviews	12,000	train 3000/3000 test 3000/3000	105,220	Weimar university [6] product reviews on Amazon
German phone calls	507	train 462/135	-	Matelso GMBH Transcribed marketing phone calls

Table 1. Datasets and their features.

3 Experiments

The main challenge in sentiment analysis on text data is representing document as vectors that can be fed to machine learning algorithms.

Analytical method

A nalytical methods of sentiment analysis are using words, words counts and other word distribution based features for representing a documents. Bayes is a very simple analytical theorem where it uses the probability distribution of current data which are classified in classes and their features to calculate the probability of a new unclassified datum(here a document) being in a specific class given it has some known features[2]. Equation 1 shows how prediction of a new sample's class given some features is performed.

$$_Y = argmax_{\{k=1..K\}} P(C_k) \prod_{i=1}^{n} P(x_i|C_k) \tag{1}$$

For implementing the Bayes algorithm we have used the word frequency as the features for each document. We adapted Bayes classifier from Scikit-learn library [7]. Also this is a very simple method it has always been very effective. even though by nature it assumes independence between features (here words) which can be counter intuitive. The reason it still functioning arguably well relies on the fact that, it is possible in short sample length with limited number of words a lot of proportionally important semantic features can be hidden in negations (ex: not the best movie) or in the relations between words and context, but when the length of sample tends to grow large, rare words and negated sentences, tend to carry proportionally less semantic meaning. The major factor which determines the success of this method is the distribution of words among classes which is the result of counting words frequency as feature. This means if there are words that are more likely to be appearing in one class and not the others, the classification will have better accuracy. In this experiment we also did a stemming preprocessing on the samples to check the effect of removing inflections and using their word stem on the results of the Bayes classifier. The assumption was: word inflection should change the distribution of words which are semantically related and accumulate the effect of different inflections of a word with semantically same polarity. the result shows stemming causes over-fitting in the process of training a Bayes classifier, figure 1.

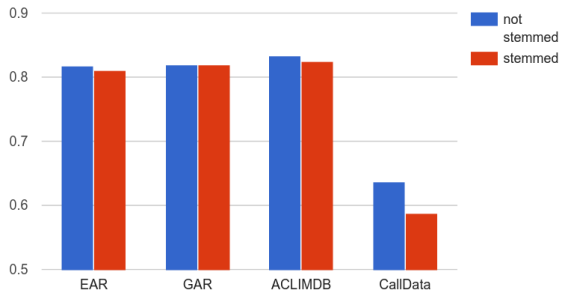

Fig. 1. accuracy of Bayes classifier.

O ne important observation in this experiment is the fact that Naive-Bayes has competitive and sometimes even better results than ANN based methods. But for the call data dataset where there seems to be not enough samples for training a Bayes classifier and the number of shared words in negative and positive training samples are considerably high, it can not compete with the other methods. Specially we have to consider the fact that Bayes method can not utilize the potential of unlabeled data. It would also have the problem of giving a probability to the new words that have never occurred in the training dataset. Also we have to consider that the input of this method for our system would come from an automated speech-to-text system which mean the possibility of having words that are completely out of language vocabulary is very very low. We can also argue the number of words that are used in conversations are much more limited than words used in a written context. We are also assuming that there should be words in conversations of one class that would never occur in the other concept for example in the context of selling products and posting it, words for describing the address would be very less likely to happen in the 'NOT_SALE' labeled item. This means that we should not completely take the Bayes algorithm off the choices and we should do another experiment with Bayes when more data is available.

ANN based methods

ANN(Artificial Neural Networks) and it's variant RNN(Recurrent Neural Networks) based methods specially LSTM have recently shown very good result in almost all the AI problems related to a sequence labeling [8] [9]. To be able to apply them, we have used the adapted code from tflearn library[10](a deep-learning library built on top of Tensorflow[11]) which is using a word embedding layer to transfer text documents into a vector representation. These vectors of words are next fed to an ANN network which at the end was connected to a soft-max layer where the positivity or negativity is decided. There are many hyper parameters that can affect the results of the classifier. Parameters like Optimizer function, Number of epochs(an epoch is a complete training pass through a given dataset), dropout (drop out is a regularization technique where some of the node outputs are randomly set to zero to prevent over-fitting). To find the best hyper parameters and best network architecture we have used several varieties of possible ANN networks. An LSTM(Long Sort Term Memory) network which is a recurrent neural network, A dynamic LSTM network which is an LSTM where the length of input vectors can differ, and a BDRNN(bidirectional recurrent neural network) where it uses the future input in a sequence as well as past input for prediction.
To find the best hyper parameters for Optimizer function, we have run the LSTM, BDLST, DLSTM on all the datasets and the results were not dependent on the dataset or the ANN architecture used. Figure 2 shows representative result of the mentioned experiment where it illustrates other not only optimizers are not competitive with adam they are on the base line of randomness. maybe prolonging the learning duration eventually lead to convergence of the mentioned methods but this would consume a lot of time and memory resource which is not suggestible.

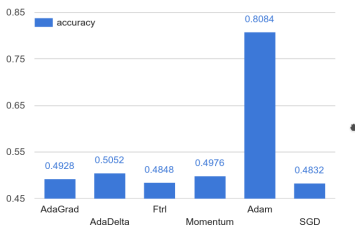

Fig. 2. LSTM accuracy on IMDB reviews dataset using different optimizers

To find the best number of epochs for the experiments we have compared the best curves on the datasets and realized that 12 epochs can be a reasonable number. Results in the figure 3 shows classier converges much earlier on aclIMDB dataset which gives us a hint about later comparison between the results of classification on datasets. One possible reason for faster convergence here can be the fact that aclIMDB reviews are all reviews about the same category, while the other datasets are reviews about three different categories.

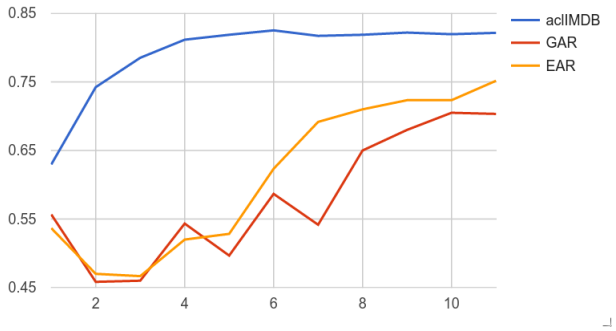

Fig. 3. Best Leaning curves for each dataset and the number of epochs used

The experiment for finding the best dropout was the one with the most unexpected results. Our assumption was the dropout should have an almost static nature were above a certain number causes data loss and reduced accuracy and bellow a certain number would cause over-fitting and thus reduced accuracy. To our surprise optimal dropout number is highly Dependant on the nature of data-set.figure 4 . One very interesting observation here was that for GAR(German amazon reviews) dataset, the best drop out was 80% which is a considerably high number. Another observation is accuracy on German dataset never reaches the accuracy on it's English counterpart which could be due to the German grammar or inflections in German language.

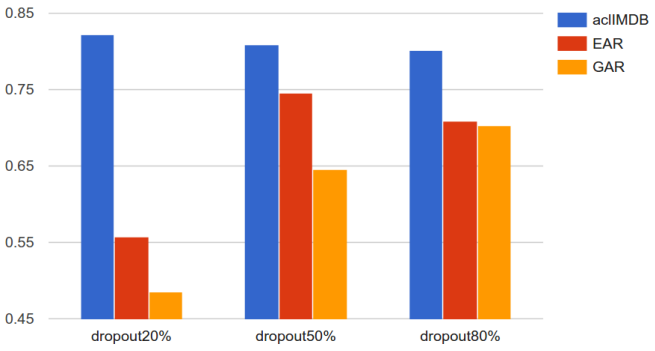

Fig. 4. effect of dropout in accuracy on different dataset

vectorized representations of Word and document

The third approach taken in this paper is based on top of embedding algorithms that are utilizing the probability of word distribution regarding their neighbors. These method are transferring words to a high dimensional space where many syntactic and semantic features are captured in the form of distance[3]. Figure 5.

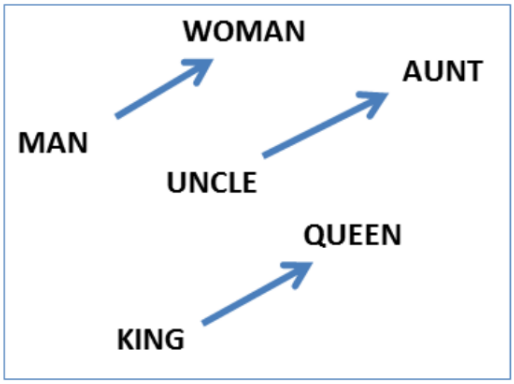

Fig. 5. Captured semantic information in the form of distance using word2vec algorithm

T his representation of words can be used to classify each word as a semantically positive or negative and can also help in sentiment classification of a document. Word2Vec[12] is one of the most known samples of this method released by google. This method can be run on text corpus in an unsupervised manner, and it would capture the semantic and syntactic relation between words. This would help the semantic classifier to use this features and apply them also for the words that never happened in the training dataset. This method can be used reliably for finding the semantic polarity of individual words but when the context is bigger then a word for example a sentence, paragraph or a document, it would be important how this individual word vectors are being used to predict a document's polarity. Gensim[13] doc2vec which is based on top of word2vec is publicly available code that we have used to combine the individual word vectors into a document vector. This algorithm is first trained on the unsupervised data and later we can get an output vector for a given input document. The output vectors of documents is later fed to a logistic regression layer and a softmax layer to decide the polarity. The network is trained with the train data for 150 epochs where the accuracy converges to a reasonable number with a small tolerance. Table 2 shows the final results of using this method which shows very competitive results.

dataset	accuracy
aclIMDB	0.8719
EAR	0.7677
GAR	0.7682
calldata	0.84

Table 2. doc2vec accuracy over different datasets

t he main advantages of this method is the possibility to do unsupervised learning for learning the word vectors which even can be done on other available data from the same language. The second advantage of this method is its fast classifier train time. Also train time for the embedding layer is time and memory exhaustive. Results shows very good output for the aclIMDB dataset which should be due to its relatively large number of available train data. The good results for call-data dataset is our main concern. It seems there are features in this dataset that doc2vec method can exploit. Finding the reasons behind the good results with doc2vec method and trying to improve it can be a part of future works.
One of the concerns that we had about the doc2vec method was, is it conserves the semantic shape of the document? by semantic shape we mean the relation between different parts of text. To confirm this we have fed the output of doc2vec to a LSTM layer and then to a softmax. We assumed if there is relation between the features extracted by doc2vec the learning curve would have a smooth increasing over time. We also wanted to see how adding an LSTM would affect the classifier's results. figure 6 shows that using an LStM layer makes the predictions worse then the previous experiment. The fluctuating values also shows that there is no sequence shape conservation in doc2vec which closes the possibility to use sequence learning approaches on top of doc2vec.

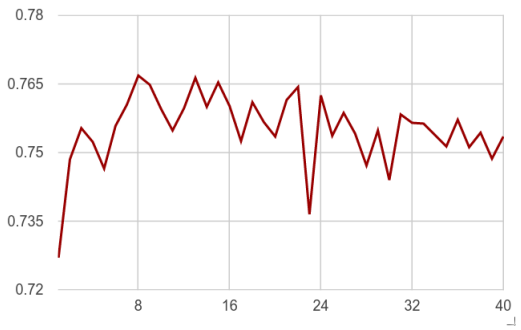

Fig. 6. Accuracy of LSTM classifier on top doc2vec input

F asttext library[4][14] which is the latest try of Facebook for getting an efficient representation of word in vector space is also another method which has been used here for comparison. Fast text library comes with an inbuilt text classifier which we have used to compare it with the out put of Doc2vec-logistic classifier. The most important advantage of facebook fasttext is, very time and memory efficient implementation of the algorithm, where training and testing time is remarkably fast. Table 3 shows the results of using fast-text algorithm for classification

dataset	accuracy
aclIMDB	0.85
EAR	0.756
GAR	0.762
calldata	0.83

Table 3. fast-text algorithm accuracy over different datasets

4 Conclusion

Transcribed data from Phone conversations in Matelso GMBH has a few main features. 1-the sample size is limited 2- conversations are transcribed using Bing automated text to speech recognition system and a percentage of tolerance for words being mistakenly transcribed should be considered. 3- the text is result of transcription of a conversation so repetition and not following correct grammar is very likely. By taking into account the special features for call data we realize the methods like LSTM or BDLSTM which are trying to use the sequential structure of data as an important feature are not performing as good as methods which are using semantic polarity of individual words and sentences. We also realize due to shortage of train dataset which would be the case for most similar experiments on voice data Bayes method would not be satisfactory enough. Best recorded result in our experiments was with Doc2vec method, and we would suggest industry or other researchers to use this method for the analysis.

5 Future works

There can be many more classes of data that are useful for the purpose of marketing and evaluation which can be used to evaluate these methods in a future work.
The voice data can also have features like tone, etc which maybe possible to be exploited in the classification. working on the voice data and extracting features from it can be a future work.

6 Acknowledgements

call-data dataset and the resources for using it and the labeling process on this data has been done with the help of MATELSO GMBH

References

1. H. Korashy W. Medhat, A. Hassan. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, 5(4):10931113, 2014.
2. Harry Zhang. The Optimality of Naive Bayes. In Valerie Barr and Zdravko Markov, editors, *FLAIRS Conference*. AAAI Press, 2004.
3. Omer Goldberg, Yoav; Levy. word2vec explained: Deriving mikolov et al.'s negative-sampling word-embedding method. *arXiv:1402.3722*, 2014.
4. Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.
5. Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.
6. Peter Prettenhofer and Benno Stein. Cross-Language Text Classification using Structural Correspondence Learning. In *48th Annual Meeting of the Association of Computational Linguistics (ACL 10)*, pages 1118–1127. Association for Computational Linguistics, jul 2010.
7. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.
8. Alex Graves. *Supervised Sequence Labelling with Recurrent Neural Networks*. PhD thesis, Technische Universitat Munchen, 2012.
9. Andrej Karpathy. The unreasonable effectiveness of recurrent neural networks, 2015.
10. Aymeric Damien et al. Tflearn, 2016.

11. Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
12. Tomas Mikolov Kai Chen Greg Corrado Jeffrey Dean. Efficient estimation of word representations in vector space. *CoRR, abs/1301.3781*, 2013.
13. Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA.
14. Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*, 2016.