Leveraging spatial distribution of affect words in Sentiment Analysis

COMP 599 - Term Project

Vincent Petrella (260467117)

***Abstract* - We propose an overview of a feature selection process applied to sentiment analysis on a particular type of language documents: movie reviews. We explored a standard dataset and measured the performance of baseline machine learning techniques employing features of increasing complexity. Furthermore, we investigate a new predictive model in the hope of leveraging the structural singularities of affect words distributions in the aforementioned types of documents.**

I. Introduction

Ever since researchers have been able to gather and process large volumes of data, notably thanks to the advent of the Internet, the Language Processing community has been actively investigating the problem of categorizing text documents. The most obvious classification task at first was topical classification, which tried to infer the subject of a document by some language analysis. Analogical to this concept of topical classification, came the problem of inferring the sentiment expressed in a text. This sentiment analysis task therefore aims at inferring whether or not a given document expresses a positive, negative or neutral sentiment. It is not to be confused with emotion analysis, which aims at inferring the particular emotion transcribed by the author (e.g. fear, anger, joy ...). The applications enabled by research in sentiment analysis are many and include for example the ability to track public opinions on various political or economical matters [1]. Another interesting application for psychological studies is to provide a framework for the automated analysis of people's mental well being from social media content [2].

In recent years, a lot of work in the Sentiment Analysis community was aimed toward analysis of Twitter text data. Twitts have the advantage of being very concise and used by most users to express an opinion, very often with a non neutral co-notation. In conjunction with the profusion of Twitter data available, twitts become an ideal case study [4]. However, such data is unlabelled, and supervised learning methods to train models on such data therefore need a significant amount of human involvement.

For this project, we elected the use of movie reviews. Reviews are most often well articulated opinionated text, punctuated most of the time by a form of rating, giving a "ground truth" indication of the sentiment developed by the author. With such easily accessible labelled data, reviews (and here movie reviews) are therefore another very practical case study for Sentiment analysis.

II. Related Work

Work in Sentiment analysis has been carried out for several years now. Early work on text sentiment classification was partially knowledge-based, using linguistic rules and heuristic based models [5]. The profusion of data available with the Internet boom allowed researchers to focus on statistical approaches to Sentiment Analysis. In 2002, B. Pang and L. Lee from Cornell University proposed a simple model that applies machine learning techniques to movie reviews classification [3]. Since then, new Natural Language Processing tools have been leveraged to offer more complex approaches to the problem of movie reviews classification. Techniques using word vector representations [6], and deep learning techniques on Tree bank compositionality [7] have been developed by researchers at Stanford University. In the broader field of Sentiment Analysis, deep learning methods have also been elaborated for analysis of short text data, such as twitts [8]. Overall, the topic of Sentiment Analysis has been and continues to be widely discussed and the state of the art remains in perpetual evolution.

III. Methods

A. Data Description

For our discussion, we elected to use the dataset proposed by B. Pang and L. Pee: the "Cornell Movie Review Data" [3]. This set contains 5006 movie reviews from 4 different authors extracted from the website *IMDB*. The reviews are several paragraphs long, and come with a subjective rating in the range from 1 to 10. Furthermore, the authors also included smaller subtexts of each review containing only the extracted "subjective" content. Finally, the authors also provide 3 and 4 classes labelling of Sentiment appreciation, based on the rating of each review.

In addition to this dataset, our work explored the use of a particular lexicon of affect words to try to leverage linguistic knowledge in our research. We used the "Max-Diff twitter lexicon" collected by Kiritchenko, S., Zhu, X., Mohammad, S. for their own publications [9]. This lexicon contains around 1500 words collected from twitts, that are accompanied with a decimal values of "sentiment intensity" ranging linearly from -1 to 1, with 0 being the neutral sentiment. Words in the lexicon also included some "hashtags". Such entries are irrelevant and thus the first character "#" was removed from each of these (e.g. "#disgusting" would be replaced by "disgusting").

B. Baseline Models

As a first step in this project, and to serve as a basis for further comparisons, we implemented simple machine learning classifiers, sharing similarities with B. Pang and L. Lee's work [3].

1) Feature Processing: For this first machine learning approach, we used a simple Bag Of Words feature representation for our movie review datapoints. Each text is tokenized into n-grams representation. The set of all the different tokens collected in the entire dataset forms our Bag of Words. A feature datapoint is then the frequency of each feature (word in the Bag of Words) for each review.

As an attempt to enhance feature selection for the classification task of Sentiment Analysis, we introduced a method to give more weight to sentiment related features in the Bag Of Words. to this effect, we multiply features in a datapoint with the "Sentiment Intensity" available in the lexicon. To assign as much weight from a strong negative sentiment (low negative value) and a

strong positive sentiment (high positive value), the range of sentiment intensity was mapped from [-1,1] to [0.1,1] by taking the absolute value and shifting 0 to 0.1. As such, an unknown "neutral" feature would still carry some weight in the classification task.

To simplify and regularize our classifying task, we cropped the feature space to only contain the $k = 2000$ most prominent features (words) by summing their values over the whole dataset. In cases where the intensity multiplier was used, this sample was taken after the features were multiplied.

2) Machine Learning Algorithm: Using the Scikit-learn python library, we implemented the following classifiers: Logistic Regression, Support Vector Machine (SVM), and the simple Naïve Bayes (NB). 2-Fold cross-validation was performed and the algorithm were run several times with different validation data.

The SVM algorithm used a linear kernel and we utilized the Gaussian Naïve Bayes implementation.

C. Leveraging spatial distribution of affect words

For the second part of this project, we developed a model to more properly account for a specificity in movie reviews. We observed that reviewers express their objective and subjective opinions at specific places in their pieces. With that in mind, it is tempting to believe that the use of affect words in some parts of a text are more pronounced that in others, and would depend on the appreciation expressed in the review. For example, a very bad and a very good movie would produce a review with respectively many bad and good affect words in the whole text; while a mixed appreciation would highlight qualities and criticisms in various areas.

In light of this hypothesis, we devised a machine learning approach to try to leverage these structural singularities based on deep Convolutional Neural Networks (CNN). In our model, a text review is seen as a time series of affect words distributions, with each time window being a contiguously sampled spatial location in the text.

1) Convolutional Neural Networks: In the convolutional layers of these types of neural networks, neurons take their inputs from spatially contiguous non-overlapping subsets of the data and are sometimes interleaved with dense, fully connected hidden layers. An example of CNN is the LeNet-5 model, which is particularly strong in the area of handwritten digits recognition. CNNs are particularly well suited to learn from data patterns laid out in time series [10].

2) Feature Processing: To obtain a meaningful representation of the distribution of affect words per text spatial locations, we divided each text into a fixed number of M text windows. For each window, we detect the distribution of sentiments by computing a vector of similarities between each word and fixed number of N affect words. These affect words were selected by taking the biggest absolute value of the "sentiment intensity" from the sentiment lexicon obtained in the first part of the project. To obtain a similarity vector between a word and all affect words, we used Google's word2vec measure of similarities between two words, pre-trained on the Google News Dataset [11]. Each word in a window is thus mapped to an N-dimensional vector, and are then averaged over the whole window to compute the sentiment distribution vector for this text window. The resulting features are thus an $N * M$ 2D matrix. The collection of all the reviews, once processed, can thus be aggregated in a 3-dimensional Tensor. The process is visually explained in Fig. 1.

3) CNN Architecture: The usual CNN architecture showcased in the literature is used for learning feature maps across contiguous 2D signals (such as images). To find an appropriate CNN architecture for learning patterns laid out in time series, we looked into the machine learning literature and derived a similar architecture as in [12]. We use a first convolutional layer over the time dimension, then across both time and sentiment distribution, and across time only again. Between the convolutional layers are intertwined subsampling layers, using the max-pooling technique.

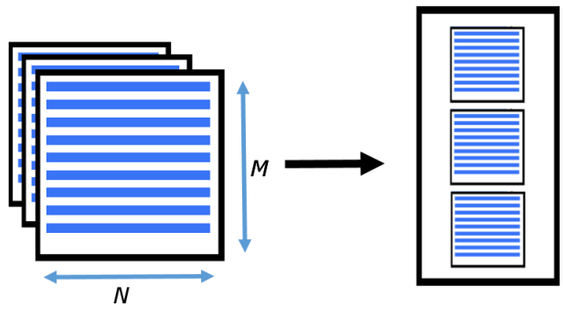

Fig. 1: Datapoints (movie reviews) are processed into $N * M$ 2D matrices, aggregated into a 3-D tensor.

4) Implementation: To implement this complex process, we used a variety of tools in Python. The data was loaded in a similar fashion as for the Baseline models, then similarities between text words and affect words were computed using the Gensim Word2Vec interface [13]. Finally, the CNN was implemented using Theano [14] with the aid of the lightweight "Lasagne" library for Theano, that simplifies building complex Neural Networks infrastructure. Theano simplifies coding machine learning algorithms and mathematical expressions using symbolic calculations, and also seamlessly translate matrix operations into CUDA kernel to easily offload learning tasks onto a GPU.

IV. Results

A. Baseline Models

We tested our best models on the Cornell Movie Reviews dataset using respectively 2, 3 and 4-classes labels. 2-Classes labels were cut off at 6/10 being a negative review, which gave a 50/50 ratio of negative and positive reviews; 3-Classes labels cutoffs were at {4,7}/10 and 4-Classes labels were cuttoff at {3,4,6}/10. We also tested with and without multiplying features by sentiment intensity (enhanced features).

Our best results for the 2-classes classification task was obtained using SVM and enhanced features, we obtained an accuracy of 72.67% and the following Confusion Matrix:

	0	1
0	1027	318
1	366	792

Accuracy without enhanced features: 72.59%

For the 3-classes classifier, we obtained a best result of 58.37% accuracy using SVM and enhanced features and the following confusion matrix:

	0	1	2
0	347	269	87
1	193	486	227
2	56	210	628

Accuracy without enhanced features: 56.93%

Finally, for the 4-classes problem, using SVM and enhanced features again, we obtained 50.30% accuracy and the following confusion matrix:

	0	1	2	3
0	152	229	167	33
1	115	244	197	62
2	49	237	412	160
3	11	70	174	191

Accuracy without enhanced features: 50.18%

B. Convolutional Neural Network

The CNN learning tasks were loaded onto a dedicated NVIDIA GTX980 GPU, which made the training process very fast. The learning process sometimes had to be restarted as the learning could get stuck at predicting the majority class, due to random weight initialization.

We report 73.35% accuracy 2-classes problem and the following confusion matrix:

	0	1
0	433	143
1	124	302

For the 3-classes problem we obtained results oscillating around 63.5% accuracy, with the following confusion matrix:

	0	1	2
0	131	100	6
1	56	230	92
2	7	104	276

Finally, for the 4-classes classification task, we obtained 55.89% accuracy and the following confusion matrix:

	0	1	2	3
0	37	58	16	0
1	26	168	112	3
2	5	81	285	37
3	0	10	94	70

More results can be found in the folder *Results/* in the submission attached to this paper.

V. Discussion

A. Baseline Models results

At first glance, we see that the improvements offered by the sentiment multiplier on the features is extremely marginal. This improvement is most likely just noise in the measurement, and the additional information added to the features does not seem to aid in the learning process. The classifiers are nonetheless quite effective, with an accuracy way above chance in all tasks.

B. CNN

The improvements offered by our complex model are also quite marginal. No improvements are seen in the 2-classes problem. This can be expected: the review of a bad movie will generally contain more negative affect words than for a good movie, no matter the distribution in the text. 3 and 4-Classes classification tasks show more promising results. The added information of the position of sentiments in the text give a boost of around 5% in accuracy over the baseline method.

However, it is crucial to put these results into perspective. A CNN requires many datapoints to be properly trained, and the size of the dataset in use was relatively small. Validating on 20% of the dataset is equivalent to training on around 1000 datapoints, a small number, that can also induce noise in accuracy measures. As we can see in Fig. 2 to 3 in the appendix, the learning process for the CNN was very unstable as a manifestation of such noise. An improvement over the baseline model could just be due to the fact that we are using a more efficient classifier, able to learn more complex boundaries than SVM with a simple linear kernel. In the case where the spatial distribution of affect words in the text does not carry information on its overall sentiment, then this Convolutional Neural Network will not fair any better than a simple Neural Network learning from aggregate of all windows. If such is the case, it is still interesting to notice that the transformation of the feature space from Bag of Words to the affect words similarities vector conserves the information about the overall sentiment of the document.

VI. Conclusion and Future Work

The hypothesis proposed in this research paper does not seem to be backed up by proper evidence. However, we can already start to pin-point a few issues with our experimental setup. The biggest downfall of this experiment is the lexicon employed, as it seems poorly adapted to the task designed. In fact, many of the most intense affect words in the lexicon were words that one would never find in a professional movie review (which is the kind of data in the Cornell Movie Review dataset), as can be showed by the table in the appendix. This may impact the performance of the proposed model, as many entries in the feature set are insignificant. It is furthermore crucial to test and train this neural network on a bigger dataset, to gain bigger confidence in the results obtained.

At this point, we cannot confirm the validity of our hypothesis. Further work would be required. A more appropriate lexicon has to be acquired. An interesting candidate would be the "Yelp Restaurant Sentiment Lexicon" [15]. Finally, a good candidate for a new movie review dataset would be Stanford's "Large Movie Reviews Dataset" [6].

VII. Statement of Contributions

We hereby state that all the work presented in this report is that of the author.

References

[1] Loren Terveen, Will Hill, Brian Amento, David Mc-Donald, and Josh Creter. "PHOAKS: A system for sharing recommendations" *Communications of the ACM*, 40(3):59–62, 1997

[2] Wang, Xinyu, et al. "A depression detection model based on sentiment analysis in micro-blog social network." *Trends and Applications in Knowledge Discovery and Data Mining. Springer Berlin Heidelberg*, 201-213, 2013.

[3] Pang, Bo, Lillian Lee, and S. Vaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10. Association for Computational Linguistics*, 2002.

[4] Mohammad, Saif M., Svetlana Kiritchenko, and Xiaodan Zhu. "NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets." *Second Joint Conference on Lexical and Computational Semantics (* SEM)*. Vol. 2. 2013.

[5] Hearst, Marti A. "Direction-based text interpretation as an information access refinement." *Text-based intelligent systems: current research and practice in information extraction and retrieval*, 257-274, 1992

[6] Maas, Andrew L., et al. "Learning word vectors for sentiment analysis." *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies. Association for Computational Linguistics*, Vol. 1, 2011.

[7] Socher, Richard, et al. "Recursive deep models for semantic compositionality over a sentiment treebank." *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*. Vol. 1631, 2013.

[8] D Santos, Cıcero Nogueira, and Maıra Gatti. "Deep convolutional neural networks for sentiment analysis of short texts." *Proceedings of the 25th International Conference on Computational Linguistics (COLING), Dublin, Ireland*, 2014.

[9] Kiritchenko, S., Zhu, X., Mohammad, S. "Sentiment Analysis of Short Informal Texts." *Journal of Artificial Intelligence Research*, 50:723-762, 2014.

[10] Y. LeCun, Y. Bengio. "Convolutional networks for images, speech, and time series." *The handbook of brain theory and neural networks*,3361, no.10, 1995.

[11] Mikolov, Tomas, Quoc V. Le, and Ilya Sutskever. "Exploiting similarities among languages for machine translation." *arXiv preprint arXiv*, 1309.4168, 2013.
[12] Zheng, Yi, et al. "Time series classification using multi-channels deep convolutional neural networks." *Web-Age Information Management. Springer International Publishing*, 298-310, 2014.
[13] Gensim: Topic libraries for human *https://radimrehurek.com/gensim/*
[14] J. Bergstra, O. Breuleux, F. Bastien, P. Lamblin, R. Pascanu, G. Desjardins, J. Turian, D. Warde-Farley and Y. Bengio. "Theano: A CPU and GPU Math Expression Compiler". *Proceedings of the Python for Scientific Computing Conference (SciPy)* June 30 - July 3, Austin, TX, 2010.
[15] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif M. Mohammad. "Detecting Aspects and Sentiment in Customer Reviews" *In Proceedings of the eigth international workshop on Semantic Evaluation Exercises (SemEval-2014)*, August 2014, Dublin, Ireland.

VIII. APPENDIX

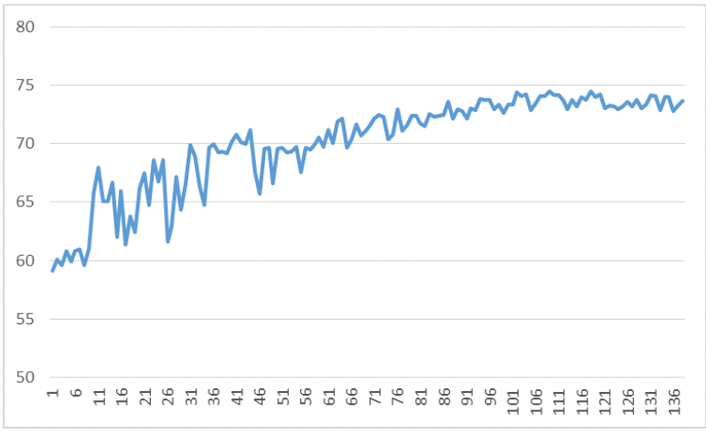

Fig. 2: Evolution of the validation accuracy over CNN training epochs for the 2-classes classification task

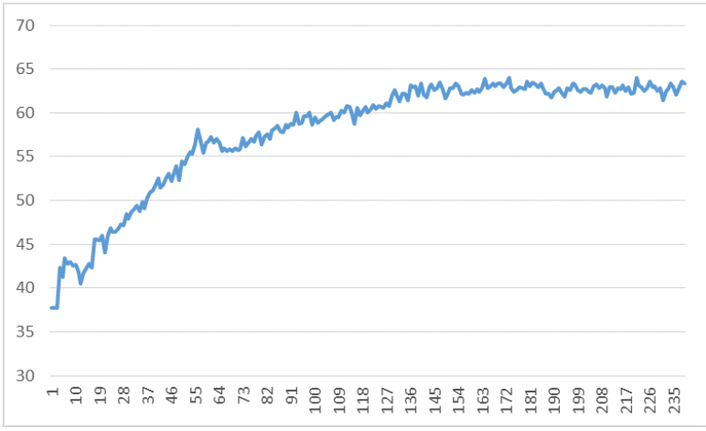

Fig. 3: Evolution of the validation accuracy over CNN training epochs for the 3-classes classification task

100 Strongest sentiment words in the Lexicon employed:

love, disappointment, inspirational, bitch, amazing, failure, kill, peaceful, killyourself, abuse, scumbag, greatness, horrid, pieceofshit, poorly, heartless, disgusted, death, fuckup, depressing, happytweet, murdered, disturbing, unacceptable, reject, spectacular, killed, notokay, fabulous, dumbass, enjoyed, died, awesomeness, disgraceful, tragedy, filth, horrendous, ripoff, angry, suck, perfection, thebest, feellikeshit, theworst, despise, sweetheart, horrible, loving, gorgeous, outstanding, liked, woooohoooo, glorious, happiness, helpful, tramp, dying, tragic, selfish, hatred, ashamed, cruel, fantastic, sickening, ewwww, inspire, despair, disgusting, die, appriciate, sophisticated, sleazy, whore, pathetic, disappointing, loved, amazzing, nasty, positive, sensational, wonderful, fuckoff, useless, magnificent, graceful, lovee, goodluck, scared, inspiration, incredible, beauty, getagrip, flawless, ugly, cretin, phenomenal, bravo, breathtaking, marvelous, putrid, yum