Comparing Approaches to Aspect Based Sentiment Analysis

Ganesh Jagadeesan and Anant Singh {cjagad2, asing45}@uic.edu

***Abstract*— The traditional sentiment analysis involves classifying the *polarity* of a given text. Aspect based sentiment analysis, as opposed to sentiment analysis, tries to classify the polarity of an entity towards a given aspect-term in the text. For example, we have a sentence "Although the battery life is not the best, the phone is pretty amazing". Here, the general polarity of the text is positive, but the sentiment towards the aspect-term "battery life" is negative. This task is difficult because a sentence can contain multiple aspect-terms with varying sentiments, even though the sentiment of the complete text could be different. In this paper, we explore few machine learning and deep learning techniques to tackle this problem.**

I. INTRODUCTION

Given a text document d_i and an aspect-term a_j, we define the problem of aspect based sentiment analysis as a classification task of determining the polarity [-1, 0, +1] of a_j in d_i. In this definition, we assume that the extraction of a_j has already been performed and is not a task for us.
We first discuss the various pre-processing steps we performed followed by the various machine learning and deep learning models employed towards this task before finally discussing some results.

II. TECHNIQUES

We will divide this section into three sections: pre-processing steps, machine learning models, and deep learning models.

A. Pre-processing Steps & Feature Engineering

We performed various feature engineering and preprocessing task, each modification to the text(specifically, formation of *nostop_text*, *combinedaspect_text*, and *noaspect_text*) was added as a respective separate column to the given csv dataset. The steps to engineer features for specific models are performed right before running that model. Following are the various steps performed.

- **Stop-word and punctuation Removal**: Removed common stop-words and punctuations from text using spaCy[2]. We call this feature *nostop_text*
- **Combine aspect-term and text**: To make the text unique, we used *nostop_text* and concatenated the aspect-term to it. We call this feature *combinedaspect_text*.
- **Remove aspect-term from text**: Another unique feature is formed by removing the corresponding aspect term from the text. For example, the text "The voice quality was amazing but the battery life is terrible." has two aspect-terms "voice quality" and "battery life". Upon performing this step, there will be two sentences formed, each after removal of one aspect-term. Therefore, first text would be "The was amazing but the battery life is terrible." and the second text would be "The voice quality was amazing but the is terrible.". Following this step, we again remove punctuations and stop-words.
- **Dependency parsing**: Using spaCy[2] we get the relevant adjectives of the aspect-terms.
- **Sequence padding**: Specifically for the LSTM-RNN[3] model, we used the default sequence padding provided by Keras[4].
- **GloVe Embeddings**: Specifically for the Memory Network[5], we use pre-trained 840B GloVe embeddings[6].
- **TF-IDF vectorization**: For the machine learning models, we converted our text sequences into tf-idf[10] bag of words using default methods provided by Scikit-Learn library[7][8].

B. Machine Learning Models

Here we provide a brief overview of all machine learning models we used. The models were trained multiple times each time using a different feature, i.e, once *combinedaspect_text*, once with *noaspect_text* and so on. One model was trained using the adjectives extracted from dependency parsing by providing them a weights based on scores extracted from various lexicons from WordNet[11]. We noticed that the best performance was achieved using just tf-idf vectorized *combinedaspect_text*. Following are the various models we tried, all of them were their corresponding scikit-learn implementations[7][8].

- **Multinomial Naive Bayes**: *Naive Bayes* predicts the class considering each feature to be independent of the other. It is perhaps the most commonly used machine learning technique for sentiment analysis tasks, which is why we decided to use this.
- **Support Vector Machines**: Feature based *support vector machines*(a.k.a SVMs) have been extremely capable of handling aspect based sentiment analysis tasks[12]. Their out-of-domain performance have outperformed most deep learning techniques. In our trials, SVMs have the best performance among all the machine learning models and it also out performs the vanilla LSTM-RNN model.
- **Random Forest Classifier**: *Random forest* classifier is an ensemble model that fits many shallow decision trees on sub-samples of the input data set. It has constantly out performed various machine learning models and in

general is a much better choice than vanilla decision trees, which is why we chose it.

C. Deep Learning Models

Here we describe the various deep learning models we implemented. For the LSTM-RNN model, we used the pad-sequence method provided by keras on *nostop_text*. For the Memory Network model, we used the pre-trained GloVe embeddings.

- **Attention based LSTM-RNN**: Initially, we tried implementing a vanilla LSTM-RNN[13] but soon switched to an attention based LSTM-RNN[3] model because of poor performance by the former. These models didn't perform well because there is no way to explicitly point at the aspect-terms in the text to somehow assert importance. It also became apparent that just pad-sequence encoding the text is not sufficient and some better representations were required. Regardless, we tried augmenting our model with 2DSpatialDroupout, BatchNormalization, EluActivation and more.
- **Deep Memory Networks**: The second deep learning model we implemented is the MemNet or the Deep Memory Network[5]. This model functions based on multiple *hops*. Each *hop* consists of an attention layer and a linear layer that get summed to become the input of the next *hop*. This model explicitly learns weights for the aspect-terms and has separate weights for the context. The inputs are given as corresponding GloVe embeddings for the context and the averaged embedding values for each word in the aspect-terms. To compensate for low computation power we used a model with 4*hops* and it performed the best among all our models, as we shall see when we discuss results.

The attention based LSTM-RNN model was built using Keras[4] and the MemNet model was implemented using TensorFlow[9]. All models were trained and validated using Statified k-fold validation. For testing, we shuffled and stratified-split the data set, saved the model and then loaded it to predict on the test set.

III. RESULTS

The data set used was a csv version of the SemEval 2016: Task 4 challenge[14]. We apply our models on the tech reviews and food reviews domains. In our experiments, **MemNet** performed best with an overall accuracy of 0.746 on the Tech Reviews and 0.762 on Food Reviews data sets in 15 epochs. We trained all models on an NVidia GTX 960M GPU with a 16GB RAM 6th Generation Intel Core i7 CPU. The attention based LSTM model seems to perform the second best, but it achieves F1-score of 0.14 on the Food reviews data set, which makes it sub-par to the feature based SVM model.

The following figures show the testing accuracy, precision, recall and the F1-score of the various models that we used on food review and the tech review data sets respectively.

Classifier	Testing Accuracy	Class	Precision	Recall	F1-Score
SVM	0.649	1	0.824	0.7356	0.7773
		0	0.3581	0.4873	0.4129
		-1	0.5073	0.51	0.5086
RANDOM FORESTS	0.713	1	0.7543	0.8743	0.8099
		0	0.4672	0.3607	0.4071
		-1	0.5789	0.4356	0.4971
NAÏVE BAYES	0.676	1	0.7791	0.8798	0.8264
		0	0.4375	0.3544	0.3916
		-1	0.6049	0.4851	0.5385
ATTENTION BASED LSTM	0.701	1	0.7527	0.9276	0.8311
		0	0.5909	0.0833	0.1461
		-1	0.5689	0.6	0.5841
MEMORY NETWORKS	0.762	1	0.7993	0.9138	0.8527
		0	0.5598	0.5022	0.5294
		-1	0.7364	0.8246	0.778

Fig. 1. Model performances on **food review** data

Classifier	Testing Accuracy	Class	Precision	Recall	F1-Score
SVM	0.672	1	0.7557	0.6978	0.7256
		0	0.495	0.4587	0.4761
		-1	0.6695	0.7536	0.7091
RANDOM FORESTS	0.695	1	0.7623	0.7234	0.7423
		0	0.625	0.3211	0.4242
		-1	0.6286	0.826	0.7139
NAÏVE BAYES	0.637	1	0.6967	0.8212	0.7539
		0	0.5522	0.3394	0.4204
		-1	0.6473	0.6473	0.6473
ATTENTION BASED LSTM	0.703	1	0.8225	0.8259	0.8242
		0	0.7083	0.3177	0.4387
		-1	0.6509	0.8426	0.7345
MEMORY NETWORKS	0.746	1	0.7936	0.7927	0.7914
		0	0.6094	0.516	0.5493
		-1	0.7584	0.8233	0.7887

Fig. 2. Model performances on **tech review** data

IV. CONCLUSIONS

It is apparent that the MemNet model makes more intuitive sense for this task which can be shown empirically as it performs the best among the methods that were used. More emerging methods[15][16] have shown increasingly better performances, even in out-of-domain testing. We would like to try out these methods in the near future.

REFERENCES

[1] Bing Liu. Web data mining: exploring hyperlinks, contents, and usage data. Springer Science & Business Media, 2007.

[2] Matthew Honnibal and Mark Johnson, "An Improved Non-monotonic Transition System for Dependency Parsing." Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Lisbon, Portugal, September 2015, pp. 1373–1378.

[3] Wang, Yequan, Minlie Huang, and Li Zhao. "Attention-based lstm for aspect-level sentiment classification." Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016.

[4] Chollet, François and others, Keras. 2015.

[5] Duyu Tang, Bing Qin, and Ting Liu, Aspect Level Sentiment Classification with Deep Memory Network, 2016, http://arxiv.org/abs/1605.08900

[6] Pennington, Jeffrey, Richard Socher, and Christopher Manning. "Glove: Global vectors for word representation." Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP). 2014.

[7] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, Gaël Varoquaux, Scikit-learn: Machine

Learning in Python, Journal of Machine Learning Research(Vol. 12), 2825–2830, 2011.

[8] Lars Buitinck, Gilles Louppe, Mathieu Blondel, Fabian Pedregosa, Andreas Mueller, Olivier Grisel, Vlad Niculae, Peter Prettenhofer, Alexandre Gramfort, Jaques Grobler, Robert Layton, Jake VanderPlas, Arnaud Joly, Brian Holt, Gaël Varoquaux, API design for machine learning software: experiences from the scikit-learn project,ECML PKDD Workshop: Languages for Data Mining and Machine Learning., 2013.

[9] Martn 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, Rafal Jozefowicz, Yangqing Jia, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Man, Mike Schuster, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Vigas, 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.

[10] Schutze, Hinrich, Christopher D. Manning, and Prabhakar Raghavan. Introduction to information retrieval. Vol. 39. Cambridge University Press, 2008.

[11] Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

[12] Svetlana Kiritchenko, Xiaodan Zhu, Colin Cherry, and Saif Mohammad. 2014. Nrc-canada-2014: Detecting aspects and sentiment in customer reviews. In Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014),pages 437442.

[13] Hochreiter, Sepp, and Jurgen Schmidhuber. Long short-term memory. Neural computation 9.8 (1997): 1735-1780.

[14] Pontiki, Maria, et al. SemEval-2016 task 5: Aspect based sentiment analysis. Proceedings of the 10th international workshop on semantic evaluation (SemEval-2016). 2016.

[15] Iñaki San Vicente, Xabier Saralegi and Rodrigo Agerri, EliXa: A Modular and Flexible ABSA Platform, CoRR, 2017

[16] Shiliang Zheng and Rui Xia, Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention, CoRR, 2018