EECS – 597 | Mid – term project report

Sentiment Analysis of Amazon Review Dataset

Introduction:

With the explosion of review websites, microblogging websites, social networks and similar platforms, sentiment analysis has become a prime objective for many organizations. Sentiment analysis plays an important role in getting feedback by being a silent listener to the chatter about the subject we are interested in. It has found applications in various domains like finance (analysis of twitter data to predict movement of stock prices), politics (to analyze twitter data from a state to predict the support), product quality (from reviews) and many other areas.

Literature Survey

There is an array of naïve algorithms and manually constructed resources to help us in understanding the sentiment. The general approach to perform sentiment analysis is to identify the representation of the text in the vector space and map it to the labelled data. Hence, we can divide the process of sentiment analysis into two parts: First is creation of feature vector space to represent the text and second is application of classification techniques to map feature space to labelled data.

The baseline method to create the feature vector space of text is to create bag-of-words representations. Another approach is to use word embedding like GloVe or word2vec to represent each word and subsequently to represent text. These representations cannot capture the relationship between the words and are created by looking at each word in isolation. For ex. Sentences like "I did not like the book, I hated it" and "I did not hate the book, I liked it" will have similar representations inspite-of having different sentiments attached with them.

Hence, there is a demand for a technique which can infer the sentiment from underlying semantics of the data without human supervision. Deep learning has proved to be a great tool in this regard. There has been lot of interest in application of neural networks towards sentiment analysis. One of the prominent neural architecture which is applied for sentiment analysis is deep neural nets and recurrent neural networks. One of such technique was proposed by Richard Socher in "Semi-Supervised Recursive Autoencoders for Predicting Sentiment Distributions". This model exploits the hierarchical structure and look at the entire sentence as whole instead of looking at words in isolation.

Overview of Dataset

We will be using the amazon review dataset available at https://snap.stanford.edu/data/web-Amazon.html . This data set contains product reviews on Amazon from May 1996 – July 2014. A snippet of the dataset for book reviews can be seen below:

helpful	summary	reviewerID	reviewText	Overall	reviewTime	asin	reviewerName	unixReviewTime
[0, 0]	Great	A2G7Y6ETCISQ4G	book.	5	07 19, 2014	60509589		1405728000
[0, 0]	Great!	A2G7Y6ETCISQ4G	ordered…	5	07 19, 2014	28740637		1405728000
[0, 0]	Learning	APRTK7SCYNOHT	nostalgic..	5	07 17, 2014	28633873		1405555200

** I have removed the words from reviewText and summary to have a better readability of the data*

Example of some of the reviewText present in the dataset are: “Story of a 10 year old who puts up with years of abuse at the hands of her mother and siblings.” “The Lord of the Rings trilogy changed my life. Its as simple as that. Everyone should read this book, and you will be changed by the reading of it. …”

The description of all the data fields is below:

- *reviewerID* - ID of the reviewer, e.g. A2SUAM1J3GNN3B
- *asin* - ID of the product, e.g. 0000013714
- *reviewerName* - name of the reviewer
- *helpful* - helpfulness rating of the review, e.g. 2/3
- *reviewText* - text of the review
- *overall* - rating of the product
- *summary* - summary of the review
- *unixReviewTime* - time of the review (unix time)
- *reviewTime* - time of the review (raw)

Information on metadata of products in also available and its various data-fields are:

- *asin* - ID of the product, e.g. 0000031852
- *title* - name of the product
- *price* - price in US dollars (at time of crawl)
- *imUrl* - url of the product image
- *related* - related products (also bought, also viewed, bought together, buy after viewing)
- *salesRank* - sales rank information
- *brand* - brand name
- *categories* - list of categories the product belongs to

Methodology

For the sentiment analysis, we are looking at just the book reviews. Also, among the available data fields, we will try to predict sentiment by using the *reviewText* field only. We first preprocessing the rating’s column. If a review issued 5 stars for a certain product, it is classified as a *positive review*. If a review issued 1 or 2 stars for a certain product, it is classified as a *negative review*. Also, the distribution of the reviewText is very skewed towards the positive reviews. Hence, we have bootstrapped from the review Dataset to create a positive:negative :: 50%:50% dataset so that the model can learn about both negative and positive reviews. Also, I am neglecting the reviews with 3 or 4 star ratings from the analysis. This creates a binary classification of the reviews.

Traditional techniques for sentiment analysis

reviewText representation as a vector:

For vector representation, we remove the stopwords and stem the words in each review. We will then choose the top 5000 most common words in the corpus and generate a simple feature vector for each review: each (stemmed) most common word will be represented as a single feature/ vector. We will then concatenate the feature vectors to create a text matrix across all reviews.

We will also use Global Vectors for Word Representations technique (GloVe) and word2vec word embeddings to represent the feature space of each word. To represent the reviews as a vector, we will then concatenate the representation of each word vector to create the text representation. We are also planning to use unigram, bi-gram, tf-idf and other embedding techniques to improve the vector representations of the reviews.

Classification Techniques:

For classification of reviews, for the baseline methods we are planning to apply techniques like gaussina naïve Bayes, multinomial naïve Bayes, Bernoulli naïve Bayes, logistic regression, linear SVM, Gaussian SVM and polynomial SVM. First, we will compare the results of these baseline methods among themselves and then look at the effect of using different word embeddings. After this, we plan to create an ensemble of all these traditional classifiers and look at the performance of this ensemble classifier.

For the second phase, I will be comparing the results of deep neural nets and recurrent networks against the traditional models we discussed above.

Results and discussion:

The analysis of traditional classification techniques on vanilla bag-of-words representation with 3000 most common words as features is shown below. For the analysis, we used 10K reviews as training dataset and 1K reviews as test dataset. Note: that we have done stratified sampling to create test and train dataset each having 50% positive sentiments.

Classification Technique	Accuracy
Gaussian naïve Bayes	67.0%
Multinomial naïve Bayes	83.9%
Bernoulli naïve Bayes	71.9%
Logistic Regression	82.9%
Linear SVM	81.9%
Gaussian SVM	81.0%
Poly SVM	79.0%
Ensemble model	84.3%
Deep neural nets (3 layers)	85.4%

Note: Current analysis is done on a corpus of 10,000 reviews. For the final report, similar analysis will be done on a large corpus ~ 100K reviews.

In addition to traditional techniques, the output for deep neural nets is also shown above.

Summary

References