Sentiment Classification of Text

Zhang, Kaili
kz2203@columbia.edu
November 8, 2013

Abstract

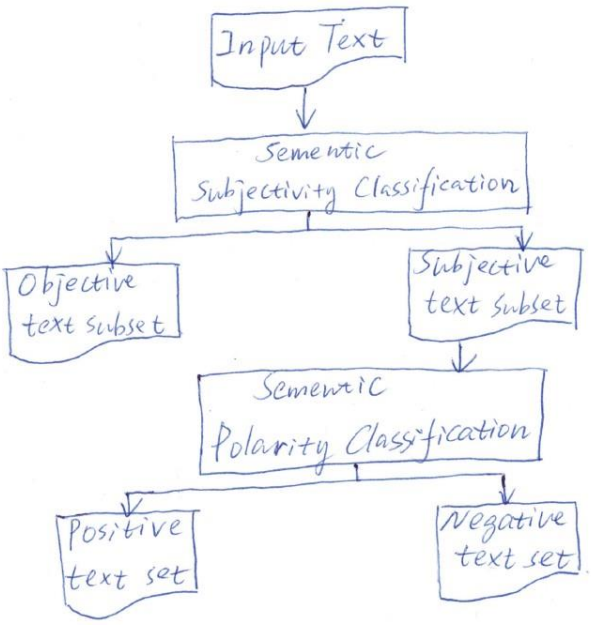
This paper presents and implements a text sentiment classification algorithms, which contains two steps, subjectivity classification and polarity classification. For subjectivity classification, 16 features are extracted, including syntactic and gram features. For polarity classification, three types of features are extracted, including syntactic, unigram, bigram, then MI and CHI feature selections are implemented. For both steps, we use a variant multinomial naïve bayes model as classifier. We use subjectivity classifier to pick out the subjective text sentence in each review as the text space for feature extraction. Finally, I got a score of 0.81145.

Keywords: sentiment classification; subjectivity; polarity; Multinomial Naïve Bayes; MI; CHI

1. Overview

A text is automatically classified as positive or negative sentiment through text sentiment classification, i.e. mining and analysis subjective information in the text, such as standpoint, view, mood, and so on. As more and more people express their viewpoints on web, text sentiment classification becomes more and more important.

For a comprehensive semantic classification, subjectivity and polarity classifications are most classic method. The following figure describes procedures of this paper.



2. Dataset

Dataset for Subjectivity Classification comes from Cornell's subjectivity datesets v1.0. Training model's training data is consisted by 2000 subjective text and 2000 objective text while testing data is consisted by 500 subjective text and 5000 objective text. For subjectivity classifier, data is Test.csv and Train.csv.

Dataset for Polarity Classification (project required part) comes from Kaggle. Four fifth of Train.csv is used as training data while one fifth as testing data. Test.csv is used for classifier.

3. Subjectivity Classification

Subjectivity Classification contains two procedures: training and classification.

In the training procedure, feature presentations for sentences are obtained from labeled training text sets via text preprocessing, text presentation; then, text subjectivity classification model is obtained via subjectivity classification model training algorithm.

In the classification procedure, feature presentations for sentences to be classified are obtained via text preprocessing, text present and feature selection; then, text subjectivity classification algorithm together with classification model is used to classify the sentences as an objective text subset and a subjective text subset.

3.1 Preprocessing

I separate data file by line into different files, each new file containing a specific review. Also, we separate each review by '.' ';' '?' '!', and then each line in review file represents a sentence in this review. Next, remove stop words. After preprocessing, each file looks like as follows.

```
1 at about 95 minutes , treasure planet maintains a brisk pace as it ra
2 however , it lacks grandeur and that epic quality often associated wi
```

(A review containing two sentence)

3.2 POS-tagging and Parsing

Part-of-Speech(POS) is criteria of type defining. In English, there are 10 different parts, like none, verb, prep, adjective. Hatzivassiloglou's research revolves that part-of-speech influence much on semantic classification. Here, I used Stanford University, NLP Groups' toolkit, an open sourced Stanford POS Tagger.

Dependency parser's function is to acquire relationships between tokens in a sentence. Input of parser is a sentence string and output is syntactic structure. Here, I used Sranford Dependency Parser.

After tagging and parsing, we have the following structure:

```
1 nothing_NN here_RB seems_VBZ as_IN funny_JJ as_IN it_PRP \
2 did_VBD in_IN analyze_VB this_DT ,_, not_RB even_RB joe_VB \
3 viterelli_NNS as_IN de_FW niro_FW 's_POS right-hand_JJ \
4 goombah_NN
```

```
root(ROOT-0, emerges-1)
prep(emerges-1, as-2)
pobj(as-2, something-3)
nsubj(emerges-1, rare-4)
det(movie-8, an-6)
nn(movie-8, issue-7)
appos(rare-4, movie-8)
nsubj(honest-12, that-9)
advmod(honest-12, so-11)
rcmod(movie-8, honest-12)
cc(honest-12, and-13)
advmod(observed-15, keenly-14)
conj(honest-12, observed-15)
mark(feel-20, that-16)
nsubj(feel-20, it-17)
aux(feel-20, does-18)
ccomp(honest-12, feel-20)
prep(feel-20, like-21)
pobj(like-21, one-22)
```

By tagging, a structure “word_POS” is gained. In parser figure, “(term_a-1, term_b-2)” means the second term term_b depends on term_a.

To run the parser and postagger, I wrote some linux shell scripts and ran it through my lab server. It spend about 25 hours to finish all the text files.

3.3 Feature Extraction

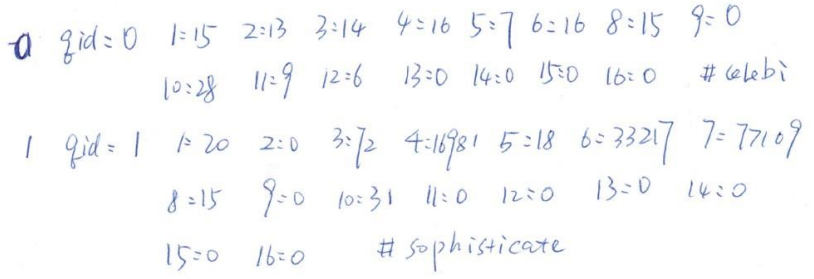
Here we extract 9 types (16 sub features) of features, Showing in the following table.

Type	Name	Description
Grams	Unigram	Present Unigram
	Adjacent unigrams	Four Adjacent unigrams, two previous, two latter
	Adjacent bigrams	Four Adjacent bigrams, two previous, two latter
Syntactic Feature	POS	POS of present unigram
	Adjacent POS	Four Adjacent unigrams’ POS
	Depending	Depending on strong or weak subjective word
	Depended	Depended by strong or weak subjective word
Lexicon Feature	Priorpolarity	Present unigram’s
	Reliability	Present unigram’s

About lexicon feature, Wilson lexicon is used. Structure of Wilson Lexicon looks like folloing figure. From this Wilson file, we could find type, prior polarity, stemmed features of some words in English.

```
type=weaksubj len=1 word1=abandoned pos1=adj stemmed1=n priorpolarity=negative
type=weaksubj len=1 word1=abandonment pos1=noun stemmed1=n priorpolarity=negative
type=weaksubj len=1 word1=abandon pos1=verb stemmed1=y priorpolarity=negative
```

Then, we formed a feature matrix.



Here, 0 and 1 represents category, qid is the number of which sentence the unigram belongs to and 1 to 16 is feature sign and feature value.

3.4 Classifier Model

Common Multinomial Naïve Bayes is chose as classifier for subjectivity classification.

3.5 Experiments and Results

In this paper, use two classic criteria Precision and Recall as evaluation criteria.

Here's the result on subjectivity testing data set.

	Objective		Subjective	
	Precision	Recall	Precision	Recall
Accuracy	87.7%	93.4%	92.3%	92.3%

After experiment, do classifier on Train.csv, Test.csv from Kaggle. Then generate subjective sentence and objective sentence for each interview.

4. Polarity Classification

In training procedure of polarity classification, a source domain labeled text set and target domain unlabeled text set are combined as training set, feature presentations for sentences of training set are obtained via text preprocessing, text presentation and MI, CHI feature selection based on pivot features; then text polarity classification model is obtained via polarity classification model training algorithm.

In the classification procedure, feature presentations for sentences to be classified are obtained from subjective text via text preprocess text present and MI, CHI feature selection based on pivot features; then text polarity classification algorithm together with classification model is used to classify the sentences as a positive sentence subset and a negative sentence subset.

4.1 Feature Extraction

Here we extract three type of features for polarity classification: Unigram, Bigram, Part-of-Speech. Bo Pang's research reveals that use trigrams as features will not increase classifier's performance, so here, we drop trigrams.

4.2 Feature Selection

I have implemented both Mutual Information(MI) and CHI Square Statistics(CHI) which

generate dictionary with relevance score for each_term. The code could be found in 'gram.py' function MI_Selection and CHI_Selection. In this part, we selected the highest 60% terms of feature matrix and removed the rest from feature matrix. After applied on the data set, it seems that Mutual Information works better than CHI Square Statistivs. About the experiment results, please refer following parts.

4.3 Classifier

Also, Multinomial Naïve Bayes is chose. However, here we use a variant multinomial Naïve Bayes which considered the frequency of term in each review. In common Naïve Bayes, this part of information is dropped. For the detailed algorithm, please refer to text book Introduction to Information Retrieval. Codes could be found in 'gram.py'.

4.4 Experiment and Results

These results are gained by applying classifier on our self generated testing set.

Accuracy	**Mutual Information**		**CHI Square Statistics**	
	Positive	Negative	Positive	Negative
100%	82.3%	81.5%	79.2%	81.1%
80%	81.9%	83.6%	79.8%	84.1%
60%	87.3%	89.1%	88.3%	84.3%
40%	85.8%	87.2%	84.9%	83.7%

Here we can see that when we choose first 60% features, we can gain a better result. Since our feature space is quite large and sparse, it is a little bit surprised me also. However, machine learning is always about parameters.

5. Discussion

By the previous method, we can also do classification on other text. However, domain knowledge is quite important. All of our training data and testing data are about movie reviews, that is to say that all of our data set come from same domain. However, if we use movie reviews training the model and then do classification on other text for example twitter text, I think the performance will be quite bad. That's the main disadvantages of this kind of supervised learning model. For semantic classification, we should design other semi-supervised or unsupervised classifer which will not be restricted by domain knowledge.

Further, since I applied quite complex method on this project, and total file containing all things need is quite large, so, I only provide some part of Polarity Classification which contains the required algorithms. For other part, please contact me if it is needed. Thanks.

Reference

[1] Cornell's Data http://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] Stanford's Postagger http://nlp.stanford.edu/downloads/tagger.shtml

[3] Stanford's Parser http://nlp.stanford.edu/downloads/lex-parser.shtml

[4] 张彦博，文本情感分类的研究，2010

[5] Text Book Introduction to Information Retrieval

[6] Peter D. Tuney Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews

[7] Sentiment Analysis of blogs by combining lexical knowledge with text classification

[8] Bo Pang, Lilian Lee, Thumbs up?: sentiment classification using Machine Learning Techniques