Review Classification and Analysis

Project Members : Srikanth Kannan
Giridara Varma Elumalai

Table of Contents

Problem Description

The objective of the project is to identify the product reviewed in the given document and perform sentimental analysis on the review. The review is categorized to one of the following product

a. Electronics
b. Food
c. Clothing
d. Automotive
e. Movies

Also, the review is analyzed for sentimental classification and is categorized to one of the following

a. Happy
b. Sad
c. Indifferent

Proposed Solution

Naïve based classifier is used to find the solution. The application is trained with a dataset which is manually labelled with product and sentiment for each review. The training of the application and analysis of the test data to find the solution is based on the extractor chosen. Following are the two different extractors used

a. Simple Extractor – Exact words in test data and training data are used
b. Advanced Feature Extractor – Various lexical, syntactic and semantic features of NLTK toolkit are used

Implementation Details

The project is implemented in Python using NLTK toolkit. The extractor to be used for a particular execution is passed as argument with value 0 indicating simple and 1 indicating advanced feature extractor. The feature set is constructed based on the extractor chosen. The path for training dataset file and testing data file are also passed as argument to the application.

I. Simple Extractor

The simple extractor is implemented for exact words in training data and test data. The training feature set and test data feature set are passed to NaiveBayesClassifier of NLTK toolkit. The classifier performs the naïve bayes classification and produces the output labels & accuracy. None of the NLP features are added in this extractor.

II. Advanced Feature Extractor

The advanced feature extractor incorporates following features of NLTK toolkit in preparing feature set of training and test data.

Lexical Feature	Syntactic Feature	Semantic Feature
Tokenization	POS Tagging	Word Sense disambiguation
Stop words removal	Chunking (Shallow parsing)	Hypernyms
Lemmatization		Hyponyms

Following are the steps done in preparing advanced feature set for training and testing data

Tokenization:

Words are tokenized using Regular Expression Tokenizer.

Filter on stop words:

Stop words are filtered from tokenized words.

Filter on POS tagging:

Words POS tagged for further use by Chunking & Word Sense Disambiguation.

Noun Phrase Chunking:

Words are filtered using noun phrase chunking or NP chunking, where the application searches for chunks corresponding to individual noun phrase.

Lemmatization:

Filtered words are lemmatized using WordNet Lemmatizer.

Word Sense disambiguation:

The lemmatized words are further filtered to find out the word which makes best sense to the context of the review. The functionality is implemented using NLTK Lesk.

Hypernyms:

Hypernyms of disambiguated word sense is found for up to two levels up and appended to feature set.

Hyponyms:

Hyponyms of disambiguated word sense is found for up to two levels down and appended to feature set.

NaïveBayesClassifier provides the accuracy based on the training feature set and test feature set. The application calculates precision and recall to compare the quality of both extractors.

Example

Training Data

happy_electronics__LOVE OUR NEW ECHO! I have been watching the reviews online and checking with friends that have purchased the Echo to see how much they liked or disliked its features. Last person I talked to went on and on about all the things there were using it for and that persuaded me it was time and Amazon Prime Day was the perfect opportunity to go for it. Amazon did a fantastic job of creating this tubular info-taining command center! There are so many cool and awesome things its able to do that I'll hit the highlights that work for our household. First, we love that it follows your voice in the room (the circle lighting will show which direction it is 'listening'), the speaker is wonderfully balanced, so whether listening to music, the news or to Alexa speaking, I have nothing but high marks for its sound quality, given its size. Next, set up (after downloading the app to our iPhones) was quick, easy and very intuitive. The more you look over the app, the more you will realize a world of 'skills' (as Amazon refers to them - we've nicked named them "echolettes" LOL) that the unit is able to perform once they are turned on and you master the right sequence of keywords to initialize them.

happy_clothing__I always buy a size 5 for Lacoste but they have switched the size 5 from a Medium to a Large. Although Lacoste says that if you wear a size 5 before the change, you should still buy a size 5, this long sleeve polo is definitely larger than past ones. Have to go to the store and try out a size 4 now to see if that is a better fit. Maybe a few runs in the dryer will shrink it enough for a better fit.

sad_clothing__France crocodile is terrible, bought a 4 t a T-shirt and large code nominal chest (in inches) 40.94, equivalent to 103.98cm, and practical for (in inches) 47.24, equivalent to 120cm. big (in inches) 6.3.. behind 3 are basically the same.

excited_clothing__The shirt is exactly what I expected from Lacoste & the service was precisely what I expect from Amazon.com

sad_automotive__Did not fit on my Buick Lucerne... even though they said they would.

angry_automotive__Okay, well they seem study and are very shiny. But be aware that while these will fit a 9/16inch stud, the lug wrench size has been reduced to 13/16 inch and not 7/8 inch which is the original size. I didn't notice that bit of information until after I tried to install them. So I can't use them because the smaller diameter means that they don't look anything like the other lug nuts. And of course you would need two sockets to change tires etc., ugh. Who knew there were so many variables when buying a lug nut? It also didn't help that the Amazon website assured me that this product would indeed fit my 2007 Dodge Durango.

angry_movies__This is an awful movie. Even by 80s standards it's possibly one of the worst, slow moving, jaw yapping so called horror films I have ever seen and I've seen some doozies.The movie is nothing but one long scene of inane boring dialogue after another with perhaps a split second of monster "scares" thrown in between. An entire hour and fifteen minutes goes by before any kind of excitement happens and it's pretty mediocre when it does. There isn't a single scare throughout, not a moment of suspense and subplots that go nowhere.

Test Data

happy_electronics__ After reading some of the other reviews with a number of issues from lack of brightness to light bleed and dead pixels, I almost feel guilty that I've enjoyed mine for nearly 2 months with flawless performance. My only guess is that the early production runs spit out a few lemons, but that's what the 1yr warranty is for during the "break in" period. Some people would rather write off a model because of a bad unit than have it taken care of and enjoy what really is a GREAT TV.

Expected Output:
Product Label: Electronics
Semantic Label: Happy

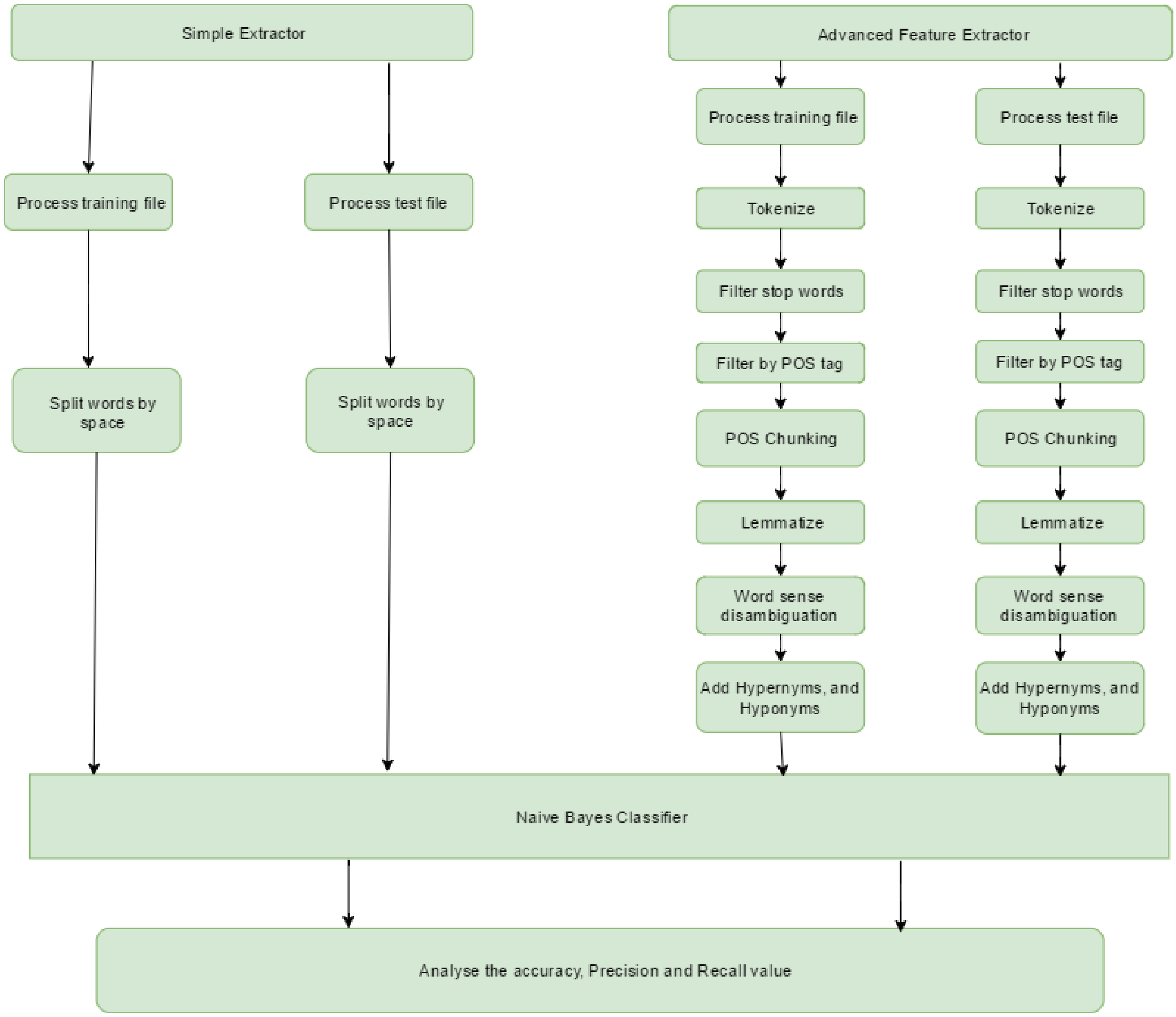
Programming Tools

Programming Language: Python 3.5.2

Third Party Tool: NLTK toolkit is used. Following are the functionalities used from the toolkit

1. NLTK Classify
2. NLTK Corpus - Wordnet and Stop words
3. POS_Tag
4. Wordnet Lemmatizer
5. NLTK Regexp Tokenizer
6. WSD Lesk

Architectural Diagram



Result

Simple Extractor Output

Review Classification

Accuracy: 0.40

LABEL	PRECISION	RECALL
automotive	-	0.00
clothing	-	0.00
electronics	0.25	1.00
food	1.0	0.33
movies	1.0	0.67

Review Sentiment

Accuracy: 0.53

LABEL	PRECISION	RECALL
happy	1.00	0.40
sad	0.45	1.00
indifferent	0.50	0.20

Review Classification

Accuracy: 0.67

Advanced Feature Extractor

LABEL	PRECISION	RECALL
automotive	1.00	0.33
clothing	0.67	0.67
electronics	0.50	1.00
food	0.50	0.33
movies	1.00	1.00

Review Sentiment

Accuracy: 0.53

LABEL	PRECISION	RECALL
happy	0.67	0.40
sad	0.50	1.00
indifferent	0.50	0.20

Summary of problems encountered

1. The required amount of related words could not be found using normal thesaurus. So, this initiated the use of wordnet.
2. Adding hypernyms and hyponyms for all words in review to the feature set, made the feature set verge large (several hundreds). This forced us to filter based on Word sense disambiguation before finding the semantics of the words.
3. The problem demanded good training set which has to be manually tagged. Unfortunately, the training data set available online is either label with product or sentiment but not both. We had to do complete manual tagging of labels in training dataset and test data.

Pending Issues

1. The application fails to analyze properly when review content contains acronyms. Acronym is not taken care.
2. If the review is vague, the application cannot find the bale of both product and sentiment. For example, if a review is just 'on a scale of 1 to 5, I give 3'. In this review, there was no hint on any product that the user reviews is mentioned nor was there any emotions shown.
3. Any new product released in the market, if it is mentioned in the review, the wordnet tags part of speech wrongly as it is unaware about the latest development in technology.

Potential Improvements

1. The application is not trained based on the emotions of the user who reviewed it. The application currently identifies the sentiment on a broad classification of Happy, Sad and Indifferent. However, the application must be improved to classify between Happy, Excited, Sad and Angry. Here the category Happy and Excited, and also Sad and Angry, are

closely coupled that the application should be wise enough to identify from the review.

2. The application currently assumes all the words in the review are spelled correctly. However, a reviewer is bound of typo errors. The application must be incorporated with spell correction functionality before analyzing the review.
3. Thematic features can be incorporated for better output. Not all reviews has explicit emotions mentioned. Some emotions are hidden and can be understood when read completely.

Observation

From the output of various test data for a single well trained dataset, we can infer the advanced feature extractor provides much accurate results than simple extractor.

1. Tokenization: In simple extractor, the application splits the words in the review by white spaces, whereas in advanced feature extractor the special characters attached to the words are properly tokenized. For example, if a review has a word *The tv quality is wonderful.* Simple extractor splits it as {The, tv, quality, is, wonderful.} whereas advanced feature extractor tokenizes it as {The, tv, quality, is, wonderful}
2. Stop words: In advanced feature extractor, stop words are filtered as they are of no use in sentimental and product classification. This step helps the feature vector more specific removing unnecessary words.
3. POS tagging and Chunking: The wordnet pos tagging and chunking helps to extract those nouns and verbs of interest which provides accurate information about the product and sentiment of the review.
4. Lemmatization: Incorporating lemmatization in advanced feature extractor, helps the feature vector to have more common words. For example, at one place review might have the word *movies* and in another place it might have the word *movie*. But both helps the application to indicate the product is Movie.
5. Word Sense disambiguation: The advance feature extractor performs word sense disambiguation which grabs the best sense of the lemmatized words increases the feature vectors weight.
6. Hypernym and Hyponym: It is observed the many words has significant number of hypernyms and hyponyms, all of which cannot be added to feature vector. Only update two levels from the current position of the hierarchy is added to feature vector which impacts the accuracy of the extractor. Considering these are hyper-parameters we observed increasing or decreasing the levels affected the results adversely.

Inference

Impact of particular feature on the accuracy of the output is analyzed for advanced feature extractor.

Feature	Product Classification (Impact on accuracy)	Sentiment Analysis (Impact on accuracy)
Stop words	Increases by 7%	Increases by 13%
Lemmatization	Decreases by 6%	Increases by 6%
POS Tagging and Chunking	Decreases by 13%	Increases by 8%
Hypernym	Increases by 7%	Decreases by 7%
Hyponym	No impact	No impact

Conclusion

The project helped to understand the role of different features (syntactic, semantic and lexical) in Natural Language Processing and how the same features affect closely related to sub problems.