Social Scoring: Movie Rating using Twitter based Sentiment Analysis

Akash Patel
Department of Computer Science and Software Engineering
Concordia University
akashkpatel1991@gmail.com

***Abstract*—Twitter's status messages consist of public views on movies, brand, business. Huge quantum of statuses are avaiable on Twitter for a particular movie but most of it are of no use as it lacks sentiment. Therefore, it is cumbersome task to go through all the reviews and obtain the overall view about a movie or brand. My movie rating system applied text mining efficiently on this corpus and calculated the overall rating using a baysian classifier. This system stands out in a way that it will not only analyze the sentiment whether it is positive or negative but it will also consider the various intensties of both the classes (positive and negative) and assign the score. I had evaluated the system using intrinsic evaluations like cross fold validation technique and manual annotation of tweets and its comparison with the system generated rating. I have also used extrinsic evaluation in which my friend was tasked to provide a rating manually after analyzing the twitter status message and comparing it with the system generated rating.**

I. Intoduction

Twitter is a micro blogging platform consisting huge volume of information which contains a variety of user reviews on different topics. The content on Twitter provides insightful thought on various brands, products, movies etc. Twitter users are growing rapidly nowadays and information from this source can be useful for sentiment analysis and opinion mining[1].

Tweets contain noisy data like acronyms, emoticons, hashtags and other special characters. Therefore, it becomes crucial to filter them out before fetching the sentiments from tweets. At the preprocessing stage, the system performs normalization to remove the URLS, special symbols and other noisy data. For example, ”http://bing.ca/img” will be converted into ” ” (a null string). Most of the tweets dont contain any sentiment. They often are mere an advertisement or an URLS. To deal with this issue, I gathered up the list of positive and negative words to detect whether a particular tweet contains any sentiment or not. The Tweets which contains sentiment(s) are then passing to the next processing stage. In addition, I am also detecting the user tag in the tweet because in my system, sometimes it is important to accumulate the sentiment of a set of users for customized rating.

Earlier research works have already covered the classification of the multi-sentenced text; like for examples, user reviews which are thoughtfully created in a proper sentential structure. However, tweets have a limit of only 140 characters. Hence, most of the time, these text pieces dont have any contextual information. Still, tweets provide necessary feedback to products or movies. I have used the multinomial Bayesian classifier to classify the tweets. In some previous works, the authors used the smileys to gather the training data for classifying tweets as a positive or negative. But, in this work, I have ”ranked” the tweets to know up to which extent movie is good or bad. I divided the keywords into five categories. For example, keywords like ”awesome” or ”great” are classified as class 5, ”good” or ”nice” as class 4, etc. Using such keywords, I designed the training data. Therefore, when a new tweet comes, Bayesian classifier will assign it to class 1 to 5.

II. Literature Surevey

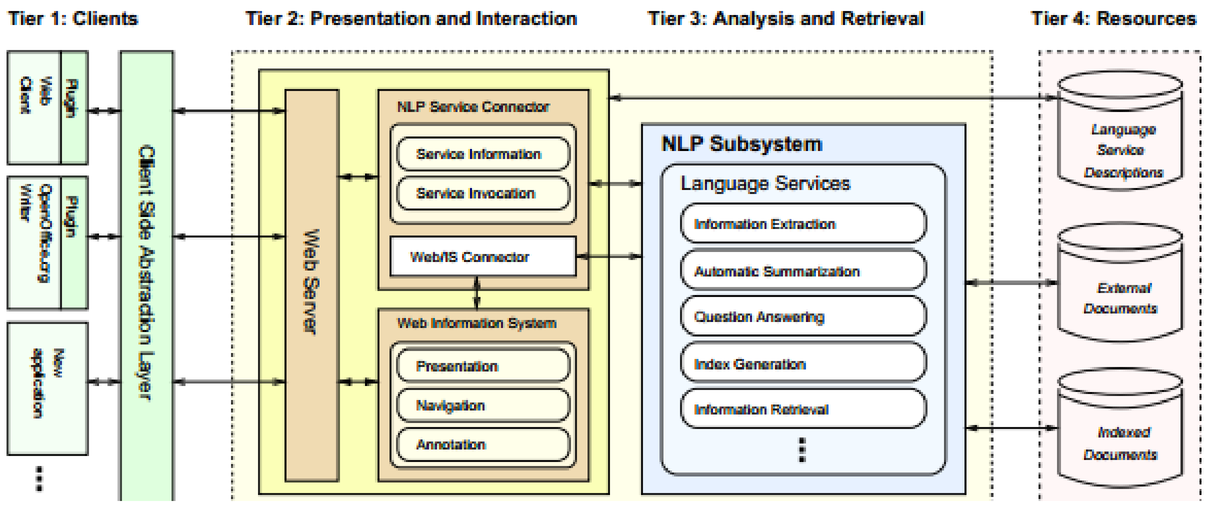
The status messages of the social networking site like twitter will be posted by users in real time. Those status messages contain sentiment regarding to particular topic. Early work on sentiment analysis has been done by Pak and Paroubek in classification of subjective versus objective. They used distance supervision technique to gather the sentiment dataset. They conclude that POS tagger and bigrams both are important[2][1]. Nathanael Chambers and Micol Marchetti-Bowick uses distance supervision learning for sentiment analysis on the political data .In the work of Nathanael Chambers and Micol Marchetti-Bowick, they have used set of keywords for gathering the data for topic identification task, and they used emoticons like :)and :-) for accumulating positive training datasets a and emoticons like :(and :-(as a negative training dataset to train distance supervision classifier [3]. Gamon performed sentiment analysis on feedback data of global service support. They concentrate more on POS tagger to extract linguistic feature. They used extensive features and concluded linguistic feature selection contributes to better accuracy in classification task[4] .

III. Background

Gate is the open source software used for text mining, while Corpus is the set of documents which can contain simple text, HTML, and RDF data[5].

A. ANNIE

A Nearly New Information Extraction System (ANNIE) is the pipeline in GATE which is composed of many components, like, Document reset, Tokenizer, sentence splitter, JAPE rules



The Semantic Assistant Architecture

etc. Few necessary components which I used in my project are: JAPE transducer, Gazetteer, Java processing resources.

ANNIE is mainly divided into two parts: Language resources and Processing resources. Language resources consist of two parts: Documents and Corpus. GATE Documents can be of any type, including XML, HTML, Plain Text, Email or PDF. Whereas Corpus is set of documents . Gazetteer list in GATE consist of a set of lst files. All lst files contains name of entities such as organizations, department names, university names, etc. And an index file called list.def is used to access these files. Therefore, during processing, processing resources uses Gazetteer list to detect the name entity in the GATE documents. When ANNIE pipeline run on the set of documents, Gazetteer processing resource generates the annotations called lookup for a matching name entity which can be used by JAPE transducer at later processing stage. [6]

B. *JAPE transducer*

JAPE is a Java Annotation Patterns Engine. JAPE provides finite state transduction over annotations based on regular expression. JAPE matches the regular expressions over the text or more complex data structure. JAPE grammar composed of a set of phases, each of which consist of pattern/action rules. The phases run sequentially and constitute a cascade of finite state transducers over annotations. The left hand side contains a regular expression. The right hand side consists of an annotation manipulation statement. LHS part is referred by means of a label in RHS part. RHS part contains JAVA code which can used for manipulation. To sum up, during the processing of documents, JAPE rule finds the matching regular expression in the text. If it finds the matching text, it fires the JAVA code of matching a regular expression[7].

C. *Media Wiki*

Media Wiki is scalable and feature rich implementation of wiki which uses PHP as a backend and MySQL for storing the information. Media Wiki has its own format to make pages[8]. Therefore, novice user of XHTML and CSS can also edit or create the pages in Media Wiki. Media Wiki does not remove the previous versions when user makes any changes in the pages, thus it allows roll back operation to the previous version. Media Wiki is extremely powerful tool for collaborative content management. It is useful for managing 1) contents 2) metadata and the 3) relation between content and metadata. But, Media Wiki doesnt support storing of the RDF triples. So, showing the connection between the entities would not be possible by Media Wiki. Therefore, semantic Media Wiki comes into picture. The data stored my Media Wiki is neither understandable nor evaluable. Semantic Media Wiki add semantic annotation or meaning.Semantic Media Wiki is the extension of Media Wiki which store the data into triple format. Therefore, query the complex data would be possible through semantic Media Wiki. Semantic Assistance: the goal of semantic assistance is to embed NLP services into client computers or web services, such as email client, web browser. To make the connection between NLP framework and client, service oriented Architecture has been made which allows integration between clients and NLP services made in GATE framework[9][10].

D. *Semantic Assistance*

Semantic Assistance: the goal of semantic assistance is to embed NLP services into client computers or web services, such as an email client, web browser. To make the connection between NLP framework and client, service oriented Architecture has been made that allow integration between clients and NLP services made in GATE framework.

Semantic Assistance framework architecture is composed of four tiers. First tier contains client side application and Client-side abstraction layer. Client side abstraction layer consists of Java classes. And the W3C web services are implemented to make the communication between client and server possible.

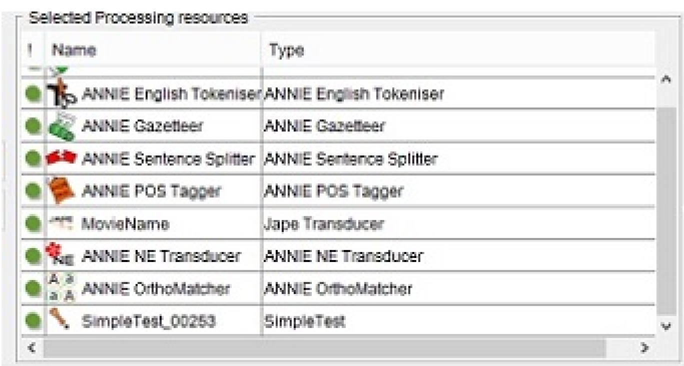
Second tier contains web server and NLP service connector. Responsibility of NLP connector is to integrate GATE NLP services. It is also in charge of querying service description, running request language service, and generating response messages. Third tier consists of an NLP system. Currently the system only supports Gate framework. It uses the GATE API to assemble the service language and to call the services according to clients' requests. Fourth tier is the resource tier, which contains Web Ontology Language (OWL) files as language service descriptions[11].

IV. Implementation

A. Pipelines

For creating the movie rating system, I have designed two pipelines which compliment each other by implementing different functionalities to complete the system. First, when user chooses the movie name, the system builds Corpus dynamically each time user want to see movie rating of a particular movie. This purpose is served by first pipeline. Secondly, I have to remove noisy tweets and classify every tweet using Bayesian classifier. This aim is accomplished by second pipeline. Both pipelines is composed of common processing resources of ANNIE including java processing resource and JAPE transducer.

In the first pipeline, MovieName-Jape Transducer is used to detect the movie name, which is entered by particular users in the clients like Media Wiki or Open Office. MovieName-Jape rule annotates the movie name with feature named movie which act as an input for later processing resources.

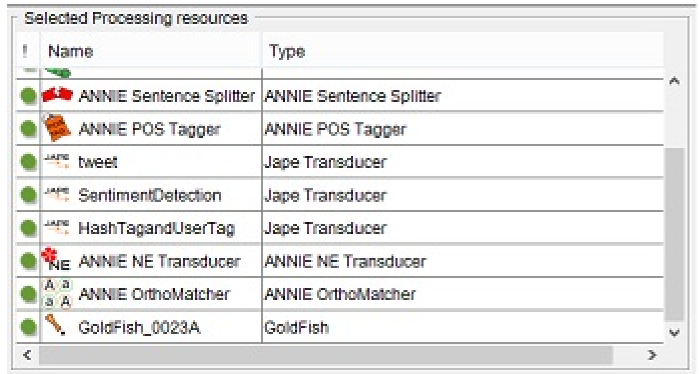


Selected Processing resources

!	Name	Type
	ANNIE English Tokeniser	ANNIE English Tokeniser
	ANNIE Gazetteer	ANNIE Gazetteer
	ANNIE Sentence Splitter	ANNIE Sentence Splitter
	ANNIE POS Tagger	ANNIE POS Tagger
	MovieName	Jape Transducer
	ANNIE NE Transducer	ANNIE NE Transducer
	ANNIE OrthoMatcher	ANNIE OrthoMatcher
	SimpleTest_00253	SimpleTest

Corpus generation pipeline(First Pipeline)

At the end of pipeline, there is a java processing resource which serves the purpose of corpus generation. twitter4j provides the simple integration of java application with the twitter services. twitter4j can be used to perform various actions like get the followers of particular user, get the list of people who is followed by which user etc. In my project, I use twitter4j to collect the status messages (called tweets) of particular movie. This resource takes the movie name as an input from the above detected JAPE rule, and then using the query class of the twitter4j library, it accumulates 500 tweets from the twitter regarding user chosen movie and generates the corpus



Selected Processing resources

!	Name	Type
	ANNIE Sentence Splitter	ANNIE Sentence Splitter
	ANNIE POS Tagger	ANNIE POS Tagger
	tweet	Jape Transducer
	SentimentDetection	Jape Transducer
	HashTagandUserTag	Jape Transducer
	ANNIE NE Transducer	ANNIE NE Transducer
	ANNIE OrthoMatcher	ANNIE OrthoMatcher
	GoldFish_0023A	GoldFish

Text mining pipeline(Second Pipeline)

which will be used for next the pipeline. Second pipeline performs real processing on the corpus which was collected in the previous step. This pipeline is made of three JAPE rules and one java processing resources. A tweet consists of many special tags like hash tag and user tag. Sometimes, hash tags contain opinions regarding particular topic, and usertags are used for referring to any twitter user. In the second pipeline, I have included first JAPE rule named HashTagUserTag which serves the purpose of annotating hashtag and usertag from the status messages. The rule for finding the hashtag and usertag is simple. We need to find the token which starts from # or '@' to find it. Here, in the pipeline, Tokenizer performs the task of converting text into set of tokens. So, we need to directly use the tokens to find whether they starts from # or '@' . Hashtag sometime contains sentiment therefore It can also be used at later stage. This pipeline consists of new two gazetteer lists. One is composed of positive words and second is composed of negative words. These two gazetteer lists are used in the jape rule called sentiment detection.

Now, we need to perform the most important task of sentiment detection. Most of Twitter contains huge volume of noisy data like advertisement, URLs etc. Therefore, finding the tweets which actually contains the actual sentiment is necessary. The JAPE rule named Sentiment Detection takes the help of two gazetteer lists described above to detect whether tweet contains opinion/sentiment or not. Sentiment detection JAPE rule compares tokens with gazetteer lists data. If it matches with positive gazetteer list data, then the system will annotate the token as a positive sentiment and if it matches with negative gazetteer list data, then the system will annotate the token as a negative sentiment. So, tweets which dont contain any sentiment will be discarded. Only those tweets which have sentiment are given to Bayesian Classifier for classification. Tweet detection JAPE rule has been used to annotate the whole tweet which is important to process at later stages.

In last, java processing resource will be exceuted. First, this processing resource does normalization on every tweet. It removes the noisy data like URL, user tags, hash tags, special character and it also discard the tweet which has a different

Rating(Class)	Keywords
5 – Rating	Amazing, Awesome, Best, Blockbuster, Breakthrough, brilliant
4 – Rating	Charming, enjoyable, entertaining, exciting, beautiful
3 – Rating	Well-balanced, amusing, faithfully
2 – Rating	Idiotic, Awful, Dislike, Criticize
1 – Rating	Bad, Worst, Hate, Failure

Keywords for extracting training dataset

language than English.

B. Bayesian classifier

I have developed Bayesian classifier for the classification of status messages in five classes and append last in the second pipeline. Here, I am giving the movie score out of five. Therefore, I have to create five different classes. For gathering the training data for each class, I manually give the rating to each keyword in the list of positive and negative words ranging from awesome - 5 rating to bad - 1 rating. Then use these keywords for generation of training data for class. Suppose, I want to generate the training data for class 1, I will use keywords like worst, bad, unlikely etc. to fetch the status messages using twitter4j, and those tweets will work as a training set for class 1. Same way training data for other classes will be generated.

Nave Bayes classifier is based on Bayesian theorem.

$$c_{map} = \arg\max_{c \in \mathbb{C}} [\log \hat{P}(c) + \sum_{1 \leq k \leq n_d} \log \hat{P}(t_k|c)].$$

Maximum likehood function

$$\hat{P}(c) = \frac{N_c}{N},$$

Prior probability

Bayesian classifier calculates the probability of every tweet with every class. And then tweets with highest probability (maximum likehood) to the particular class would be classified to that class. I have used Multinomial Bayesian classifier with uni-gram feature. Therefore, this classification algorithm doesnt classify the status message like I dont like this movie.

$$\hat{P}(t|c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}},$$

Conditional probability

So, I used negation handling approach. In this approach, the system checks for the verb, if auxiliary verb is negation then the word follows it, if it will be the positive then sentence will become negative and on the other hand, if auxiliary verb is negation and word is negative, then it convey positive meaning regarding topic.

C. OWL files

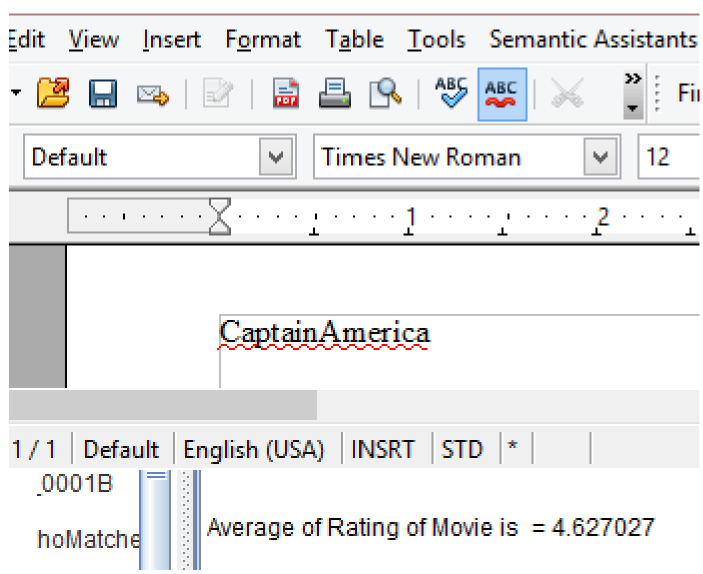
Owl file is ontology based service description. The annotations which you need in the output we have to define those annotations and feature into the owl file. I have created three different owl file. First OWL file is used for first pipeline used for corpus generation. Usage of second pipeline is to annotate tweet, tweet contains sentiment and rating of tweets. Third OWL file is doing the work of combining pipelines defined above.

V. Application

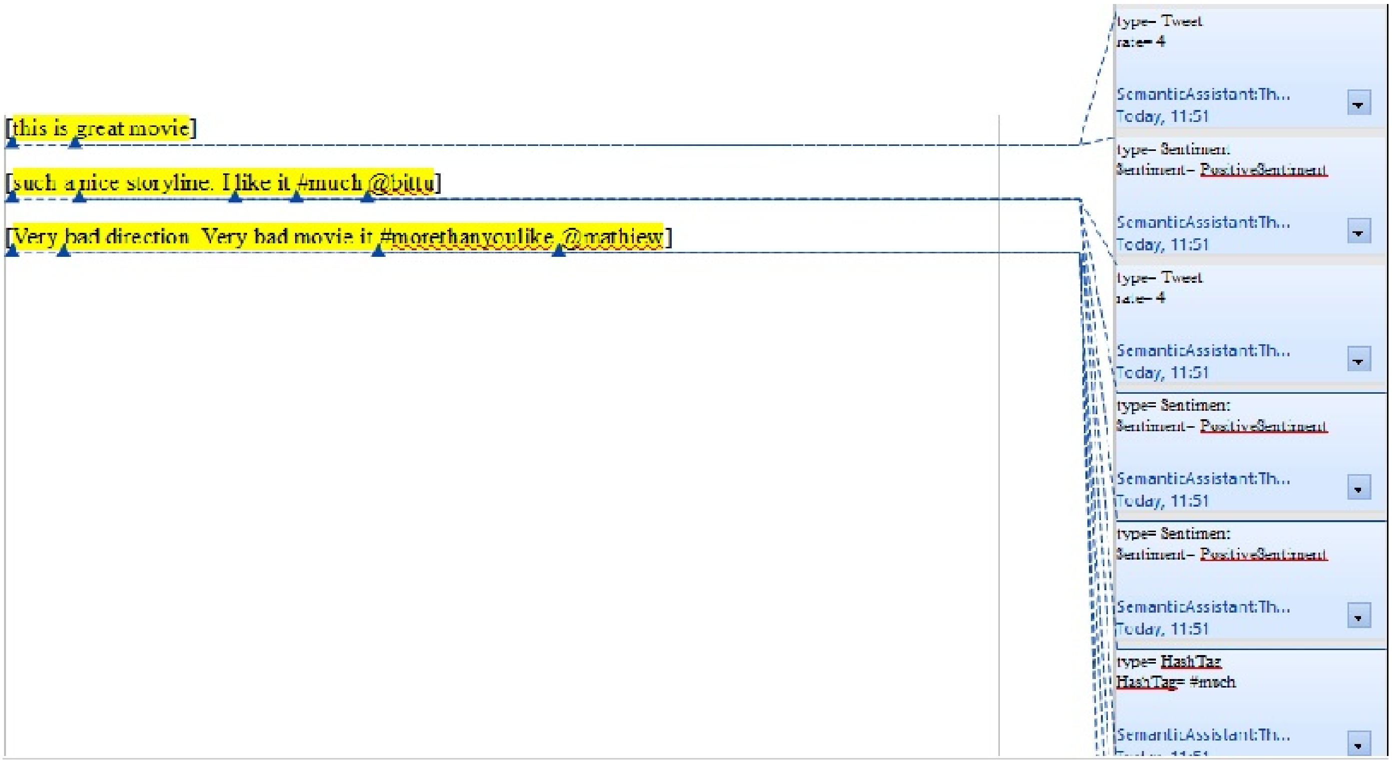
Demonstration of movie rating system was performed on two different client named Open Office and Media Wiki.

A. Open office

Wiki-NLP has the simple integration with Open Office. We need to configure the setting only in one file named semantic Assistance.xml and we can see new tab called semantic Assistance in open office which shows all the available NLP pipelines in the server.



CaptainAmerica Movie Rating(open office- First Case)



Openoffice client executes second pipeline(Second Case)

In this paper, I show the experiment with two different pipelines. One with combined corpus generation pipeline and tweet classification pipeline, and other with only tweets classification pipeline. As an input to combined pipeline is only movie name. And second pipeline take tweets as an input.

First we have to start server which deployed NLP pipeline. In First case as shown in figure, I have given the movie name Captain America in open office, then select semantic assistance tab and select combined pipeline. Combined pipeline is the aggregation of two pipelines. So, when first pipeline executes it gather all tweets related to movie Captain America and make the corpus. Now, this corpus will pass to second pipeline, and second pipeline will do pre-processing first like remove the junk characters, URL, user tags etc. After normalization, it will forward tweet to Bayesian classifier for classification.

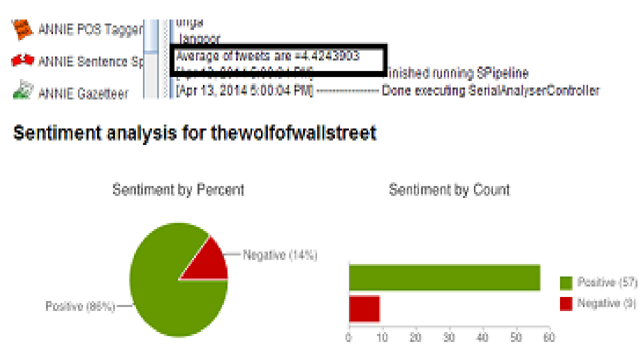
In Second case, I used the status messages as an input text and run only one pipeline to it. The Owl file of the second pipeline is used for annotations which are required for display. Second pipeline process the tweet for classification. And annotation likes tweets, Sentiment Detection and User tag and Hashtag will be displayed in open office.

B. Media Wiki

In this section, I have also tried media wiki with semantic assistance. Media wiki is powerful collaborative content management system. I am using two media wiki extensions: semantic media wiki and Semantic assistance. For experimentation, I have created a page with name tweets which contains the set to status messages and I run the pipeline on that page for displaying the annotations and background working. Final annotations are generated in tabular forms after final processing.

VI. Evaluation

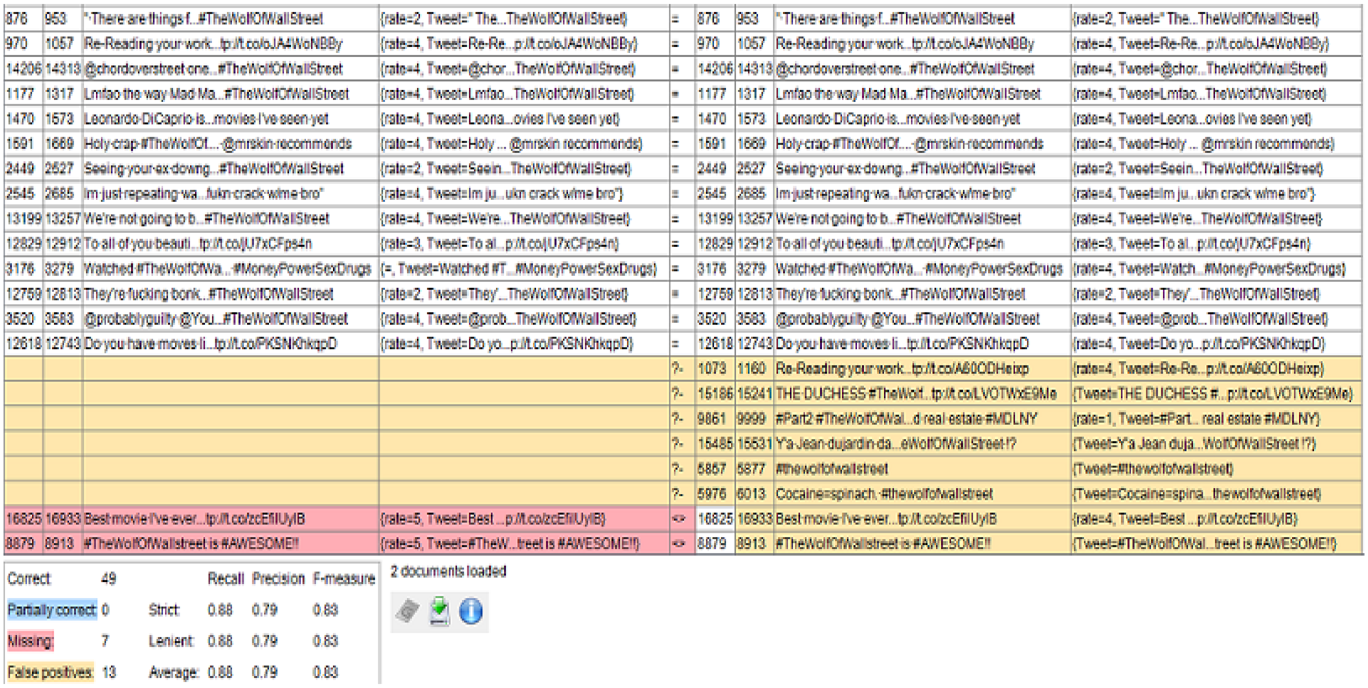
For evaluating the movie rating system, I evaluated system with three different approaches.



Comparision with sentiment 140 (Evaluation)

A. Comparison Movie rating system with online site sentiment140

Sentiment140 is site which allows discovering the opinion of people regarding brand, product, and movie etc. Senti-

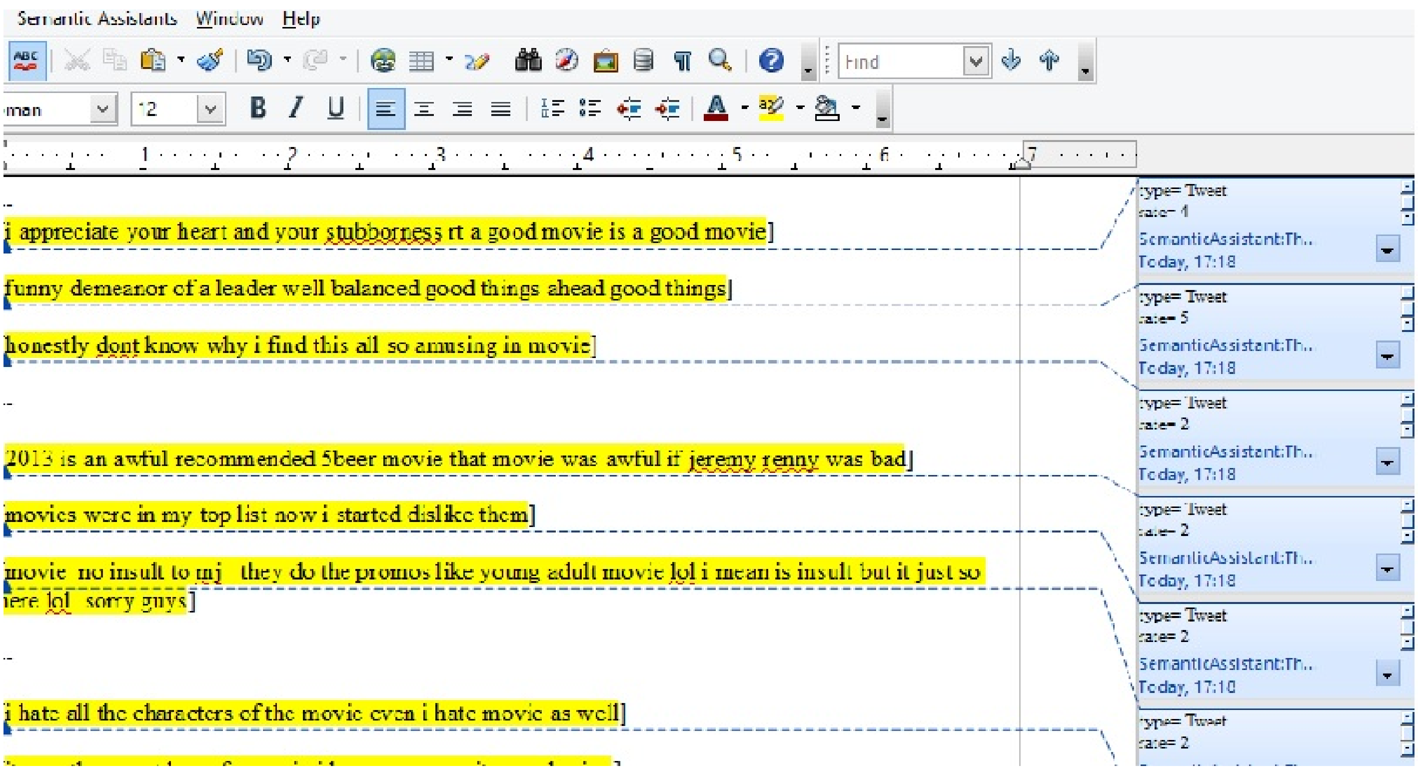


876	953	"·There·are·things·f...#TheWolfOfWallStreet	{rate=2, Tweet=" The...TheWolfOfWallStreet}	=	876	953	"·There·are·things·f...#TheWolfOfWallStreet	{rate=2, Tweet=" The...TheWolfOfWallStreet}
970	1057	Re-Reading·your·work...tp://t.co/oJA4WoNBBy	{rate=4, Tweet=Re-Re...p://t.co/oJA4WoNBBy}	=	970	1057	Re-Reading·your·work...tp://t.co/oJA4WoNBBy	{rate=4, Tweet=Re-Re...p://t.co/oJA4WoNBBy}
14206	14313	@chordoverstreet·one...#TheWolfOfWallStreet	{rate=4, Tweet=@chor...TheWolfOfWallStreet}	=	14206	14313	@chordoverstreet·one...#TheWolfOfWallStreet	{rate=4, Tweet=@chor...TheWolfOfWallStreet}
1177	1317	Lmfao·the·way·Mad·Ma...#TheWolfOfWallStreet	{rate=4, Tweet=Lmfao...TheWolfOfWallStreet}	=	1177	1317	Lmfao·the·way·Mad·Ma...#TheWolfOfWallStreet	{rate=4, Tweet=Lmfao...TheWolfOfWallStreet}
1470	1573	Leonardo·DiCaprio·is...movies·I've·seen·yet	{rate=4, Tweet=Leona...ovies I've seen yet}	=	1470	1573	Leonardo·DiCaprio·is...movies·I've·seen·yet	{rate=4, Tweet=Leona...ovies I've seen yet}
1591	1669	Holy·crap·#TheWolfOf...·@mrskin·recommends	{rate=4, Tweet=Holy ... @mrskin recommends}	=	1591	1669	Holy·crap·#TheWolfOf...·@mrskin·recommends	{rate=4, Tweet=Holy ... @mrskin recommends}
2449	2527	Seeing·your·ex·downg...#TheWolfOfWallStreet	{rate=2, Tweet=Seein...TheWolfOfWallStreet}	=	2449	2527	Seeing·your·ex·downg...#TheWolfOfWallStreet	{rate=2, Tweet=Seein...TheWolfOfWallStreet}
2545	2685	Im·just·repeating·wa...fukn·crack·w/me·bro"	{rate=4, Tweet=Im ju...ukn crack w/me bro"}	=	2545	2685	Im·just·repeating·wa...fukn·crack·w/me·bro"	{rate=4, Tweet=Im ju...ukn crack w/me bro"}
13199	13257	We're·not·going·to·b...#TheWolfOfWallStreet	{rate=4, Tweet=We're...TheWolfOfWallStreet}	=	13199	13257	We're·not·going·to·b...#TheWolfOfWallStreet	{rate=4, Tweet=We're...TheWolfOfWallStreet}
12829	12912	To·all·of·you·beauti...tp://t.co/jU7xCFps4n	{rate=3, Tweet=To al...p://t.co/jU7xCFps4n}	=	12829	12912	To·all·of·you·beauti...tp://t.co/jU7xCFps4n	{rate=3, Tweet=To al...p://t.co/jU7xCFps4n}
3176	3279	Watched·#TheWolfOfWa...·#MoneyPowerSexDrugs	{=, Tweet=Watched #T...#MoneyPowerSexDrugs}	=	3176	3279	Watched·#TheWolfOfWa...·#MoneyPowerSexDrugs	{rate=4, Tweet=Watch...#MoneyPowerSexDrugs}
12759	12813	They're·fucking·bonk...#TheWolfOfWallStreet	{rate=2, Tweet=They'...TheWolfOfWallStreet}	=	12759	12813	They're·fucking·bonk...#TheWolfOfWallStreet	{rate=2, Tweet=They'...TheWolfOfWallStreet}
3520	3583	@probablyguilty·@You...#TheWolfOfWallStreet	{rate=4, Tweet=@prob...TheWolfOfWallStreet}	=	3520	3583	@probablyguilty·@You...#TheWolfOfWallStreet	{rate=4, Tweet=@prob...TheWolfOfWallStreet}
12618	12743	Do·you·have·moves·li...tp://t.co/PKSNKhkqpD	{rate=4, Tweet=Do yo...p://t.co/PKSNKhkqpD}	=	12618	12743	Do·you·have·moves·li...tp://t.co/PKSNKhkqpD	{rate=4, Tweet=Do yo...p://t.co/PKSNKhkqpD}
				?-	1073	1160	Re-Reading·your·work...tp://t.co/A60ODHeixp	{rate=4, Tweet=Re-Re...p://t.co/A60ODHeixp}
				?-	15186	15241	THE·DUCHESS·#TheWolf...tp://t.co/LVOTWxE9Me	{Tweet=THE DUCHESS #...p://t.co/LVOTWxE9Me}
				?-	9861	9999	#Part2·#TheWolfOfWal...d·real·estate·#MDLNY	{rate=1, Tweet=#Part... real estate #MDLNY}
				?-	15485	15531	Y'a·Jean·dujardin·da...eWolfOfWallStreet·!?	{Tweet=Y'a Jean duja...WolfOfWallStreet !?}
				?-	5857	5877	#thewolfofwallstreet	{Tweet=#thewolfofwallstreet}
				?-	5976	6013	Cocaine=spinach.·#thewolfofwallstreet	{Tweet=Cocaine=spina...thewolfofwallstreet}
16825	16933	Best·movie·I've·ever...tp://t.co/zcEfIUyIB	{rate=5, Tweet=Best ...p://t.co/zcEfIUyIB}	<>	16825	16933	Best·movie·I've·ever...tp://t.co/zcEfIUyIB	{rate=4, Tweet=Best ...p://t.co/zcEfIUyIB}
8879	8913	#TheWolfOfWallstreet·is·#AWESOME!!	{rate=5, Tweet=#TheW...treet is #AWESOME!!}	<>	8879	8913	#TheWolfOfWallstreet·is·#AWESOME!!	{Tweet=#TheWolfOfWal...treet is #AWESOME!!}

Correct	49		Recall	Precision	F-measure
Partially correct	0	Strict:	0.88	0.79	0.83
Missing	7	Lenient:	0.88	0.79	0.83
False positives	13	Average:	0.88	0.79	0.83

2 documents loaded

Manual Annotation(Evaluation)



Cross Validation (Evaluation)

ment140 was created by three Stanford graduate students. Instead of using keyword for classification which has lower recall, It uses machine learning algorithm for the classification of the status messages and give you final summery.

Here, I compare my movie rating with sentiment140. I applied my both pipeline on movie named The Wolf of wall streets to get the overall rating of the movie according to tweets and I use sentiment140 to get the statistic of positivity and negativity. Comparison shows nearly the same results.

B. Manual Annotation

I accumulate the 142 tweets of movie The Wolf of wall street using Twitter 4 js Query class. Always few tweets actually give the opinion about movies. Others are just advertisement or movie dialogs. I annotate the tweets which actually contain sentiments or opinion. 62 tweets out of 142 tweets were talking about the movie. Therefore, I annotated only 62 tweets. After, I applied my movie rating system on set of 142 tweets. Total 7 tweets have been missed by my system because sentiment has been described ambiguously. 13 tweets were categorized as false positive as shown in figure.

Other 49 tweets rated well by the system. Sometime, system assign 4 rating instead of 5 rating to the tweets like ”This is great movie” but it doesnt impact on overall system because those tweets has been classified as a strong positive at least. To sum up, the recall of movie rating system is 0.88. Precision of the systzem is 0.79 and F-measure of the system is 0.83.

C. Cross fold validation

For cross fold validation, I have considered 90 tweets. 6 fold validation has been performed for evaluating system. 15 tweets for 5 class and 15 tweets used for testing I calculated average after 6 times shuffling the tweets. The accuracy or precision found out is 73 %. Therefore, it can be concluded that nave Bayesian classification algorithm provide enough efficiency for the classification of status messages. Below you can see the screenshot depicts assigned classes to tweets by Nave Bayesian algorithm during the experimentation.

VII. Conclusion

Most of the papers on movie rating system based on sentiment analysis focuses on three classes positive, negative and neutral. If the sentiment analysis has been performed for these classes, we will be able to know whether a particular movie is good or not. But, in this project I have classified the tweets in the interval of 1 to 5 according to intensity of the statements. For instance the word Awesome and Good will be classified as a positive but, here, Awesome has more positive strength than good. I have evaluated the system by two ways. First, I have manually annotated the tweets and checked the accuracy of algorithm against it. Second, I have performed cross validation evaluation to test the training data and algorithm.

For future work, I would like to make some improvements in the system. For example, sometimes people express their feeling about a movie using its dialogues. Therefore, I would also like to consider movie dialogues in text analysis task using sophisticated linguistic features of POS tagger and tackle many challenges that might arise during the normalization of tweets.

References

[1] Apoorv Agarwal, Boyi Xie, *Sentiment Analysis of Twitter Data*.

[2] Alexander Pak, *Twitter as a Corpus for Sentiment Analysis and Opinion Mining*.

[3] Micol Marchetti-Bowick, *Learning for Microblogs with Distant Supervision: Political Forecasting with Twitter*.

[4] M. Gamon., *Sentiment classification on customer feedback data: noisy data, large feature vectors and the role of linguistic analysis*.

[5] Gate, *(https://gate.ac.uk/)*.

[6] ANNIE, *(http://gate.ac.uk/sale/tao/splitch6.html)*.

[7] JAPE, *(https://gate.ac.uk/sale/tao/splitch8.html)*.

[8] Hendry Muljadi, Hideaki Takeda and A. Fujiyama, *Semantic mediawiki: a user-oriented system for integrated content and metadata management system*.

[9] MediaWiki, *(http://www.mediawiki.org/wiki/MediaWiki)*.

[10] S. M. Wiki, *(http://semantimyc-mediawiki.org/)*.

[11] Rene Witte, Thomas Gitzinger and B. sateli, *(http://assistant.cs.concordia.ca:8080/job/Semantic Assistants/lastSuccessfulBuild/artifact/semantic-assist/Doc/semassist.pdf)*.