

# Sentiment Analysis in the IT Domain an Enhanced Approach to VADER Sentiment

Kiran Hegde. Architect, Aarush Gupta. *IT Engineer*, Aparna George. *IT Engineer* and Anudeep Dhonde. *IT Analyst*

**Abstract**—Applying sentiment analysis in the IT domain requires several enhanced techniques over the conventional methods used for social media analysis, mainly due to the technical data involved as well as the purpose of the analysis. We take a deep-dive into these challenges and present a comparison of the results obtained over the two techniques (VADER analysis and our enhanced approach) along-with the interpretation of the enhanced results obtained through our approach in this paper. We will also give a description of the future enhancements to be implemented as a part of our current approach.

**Index Terms**—data analytics, IT, natural language processing, opinion mining, sentiment analysis, VADER sentiment, work logs, Remedy, Service Now, ITIL, Operations

## 1 INTRODUCTION

Sentiment Analysis or Opinion Mining refers to the use of natural language processing, text analysis or machine learning techniques to systematically identify, extract, quantify and characterize the sentiment content or polarity of a text unit on a specific entity, like an individual, an event or a product. The text could be a variety of materials including customer feedback, product reviews, survey responses or data pertaining to social media, which is analyzed and then categorized as positive, negative or neutral. The applications of these techniques include customer service and satisfaction tracking, product marketing, market forecast, gaining insights into the performance of one's own or competitor's products by a variety of industries.

We used one of the most common models, VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis, which is a lexicon and rule-based model that is specifically attuned to the sentiments expressed in social media. Empirically validated by multiple independent human judges, VADER incorporates a gold-standard sentiment lexicon that is especially suitable to microblog-like contexts. The VADER sentiment lexicon is sensitive to both the polarity and the intensity of sentiments expressed in social media contexts, and is generally applicable to sentiment analysis in other domains.

However, the data we were interested in analyzing was IT customer support work logs that would help to measure the customer satisfaction levels with the support experience provided. Upon using the Vader model, we encountered several challenges due to our data set being specific to the IT domain. This paper explains the approach we used to address those issues and illustrates the improvement in prediction accuracy we were able to achieve.

Further, we also present a few future enhancement possibilities on this technique including usage of a convolutional neural network model for sentiment analysis.

## 2 PROBLEM STATEMENT

From an IT standpoint, customers, when facing issues with software, hardware components or Functional areas, raise incident tickets with the concerned team to have their issue resolved. However, the resolution of the issue might be delayed due to multiple reasons or the solution itself might be deemed unsatisfactory by the customer in some cases, thereby making it improperly resolved, who then responds to the case with negative comments. These back and forth communications between the incident assignee and the customer are recorded as

incident work logs. Our objective is to identify the sentiments expressed in these incident work logs and identify the customer satisfaction levels with the overall incident resolution process and providing each incident ticket with a sentiment score.

### **3 EARLIER APPROACH AND CHALLENGES**

#### **3.1 VADER Approach**

This section describes the VADER approach in detail. First, the work logs are broken down into sentences and then cleaned by removing the stop words and salutations followed by stemming. Stop words are a set of commonly used words in any language. The reason why stop words are critical to many applications is that, if we remove the words that are very commonly used in a given language, we can focus on the important words instead, thus getting rid of redundant words in the process. Removing stop words reduces the dimensionality of term space. Stemming is the process of reducing a word to its stem or root form. For example, the words connect, connected, connecting, connections can all be stemmed down to “connect”. The purpose of this method is to remove suffixes, in order to have accurately matching stems, to save time and memory space. Stemming is based either on linguistic dictionaries or on algorithms. Once the preliminary data pre-processing steps are applied to the data, every sentence is passed to the VADER Sentiment Analyzer function to extract the score and polarity of each word in the sentence. The result is then used to identify top negative words in the analyzed worklogs.

#### **3.2 Challenges**

The VADER approach when applied to the IT data set that we were dealing with resulted in lesser accuracies, and often times incorrect sentiment results. Few of the challenges that we faced are explained in detail in this section. IT tickets are broad in coverage (e.g. infrastructure, software, architecture, design etc.) and very commonly involves words like errors, issues, problems, failures, etc. The foremost challenge is to automatically distinguish the factual information, which is intrinsically negative (e.g. error/issue description), from the sentiment embedded in the description. Although positive sentiments are expressed by the customer, due to the presence of commonly used words like error, issue etc., a negative sentiment is reflected. For IT tickets, it is necessary to distinguish between the objective report of a problem,

and the embedded sentiment. By design, generic sentiment analysis solutions do not care for such distinction, and thus do not yield consistent results, which is the problem we faced while incorporating VADER sentiment approach for the same. Most words, which are recurring in the social media data set, do not even appear in the IT ticket work logs - hence the scale of the range of sentiments and the assigned value to each should also be revisited and corrected accordingly.

### **4 CURRENT APPROACH AND METHODOLOGY**

This section will explain the solutions to the challenges explained in the previous section and the approach we took to maximize the efficiency for the same. Our approach to solve this problem comprises of two main steps:

1. Creation of a domain dictionary that consists of words and phrases with the sentiment related to the IT Domain.
2. Sentiment analysis process that assigns polarity scores to ticket words and phrases, and aggregates individual scores for calculating the overall ticket polarity.

The domain dictionary creation process involves three major steps:

#### **4.1 Pre-processing the seeds**

It consists of selecting and preparing the seeds for creating the dictionary. The seeds are words or expressions extracted from IT tickets, which are prepared by manually assigning a polarity orientation to the seed according to the domain (i.e. “horrible” is negative and “excellent” is positive) and classifying each term candidate for expansion according to one of the following POS (parts of speech) tags: “a” (adjective), “n” (noun), “r” (adverb) or “v” (verb) and appending it to the token using the special mark “#” (e.g. “excellent#a”, “extremely#r” and slow#a”). Each entry in the Domain Dictionary is uniquely defined by <token>#<POS>. In order to select the seed words, 42,372 customer worklogs were used and the most common adjectives and adverbs were formed as seeds. 61 seed words in total are used (33 Negative and 28 Positive). POS also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as well as its context – i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph.

## 4.2 Seed Expansion

Once the initial seeds are decided, we identify synonyms for these with same POS (parts of speech). Not all the tokens returned by seed expansion are relevant. When a synonym token is extracted, its polarity is also extracted. The tokens are discarded if any one of these conditions are met:

1. The token is not present in the sentiment lexicon or has a polarity score equal to 0.
2. The expanded token has an inverse polarity compared to the original seed (e.g. “slowly#r” is a negative seed, but the synonym “easy#r” is positive, which references the issue of opposite polarities.
3. The expanded token has a polarity score between -0.1 and 0.1, which excludes terms with very weak sentiment.

For each word in this dictionary, synonyms and polarity scores are extracted from SentiWordNet online dictionary and rules are used to grow the dictionary. This dictionary has 223 words and scoring is done after one iteration and is used for scoring the worklogs. Since seed words do not have any scores associated with them, we calculate the average score of negatively tagged words and assign it as score for negative seed words. The same process is repeated for positive words as well.

## 4.3 Pruning

The last step in this process is pruning. At each depth, the number of tokens in the dictionary are checked for presence in our tickets corpus (percentage of tokens present). If there is no significant increase in percentage at depths  $n$  and  $n+1$ , we can prune the tree to depth  $n$  and finalize the dictionary. In our case, we are using the depth of one.

## 4.4 Processing and Scoring

As explained in the problem statement, we are dealing with customer worklogs or incident tickets. Before we assign a final sentiment score to the incident, the following steps are performed:

### 4.4.1 Cleaning the Worklogs

In addition to the standard data cleaning steps explained in the approach earlier, since these worklogs at times contain email chains, salutations etc., which can bias the sentiment, some additional cleaning needs to be performed by removing these

salutations, signature lines, email chains, standard footers, reassignment logs, etc.

### 4.4.2 Using Domain Dictionary to assign scores

The work log is tokenized and lemma and POS (parts of speech) are extracted using spacy (python library). Lemma is the canonical form, dictionary form, or citation form of a set of words. For example - run, runs, ran and running are all forms of the same lexeme, with run as the lemma. Each lemma with POS is used to lookup for a score in the domain dictionary and is mapped accordingly.

### 4.4.2 Apply Heuristics

After assigning, the score to each word in the worklog, two major heuristics are applied:

#### 4.4.2.a Negation Identification

If negation is present in a sentence, related adjectives/adverbs scores are changed to the opposite sign. Example: “Your team is NOT effective” (Although effective has a positive sentiment it will be changed to negative because of presence of negation (NOT)).

#### 4.4.2.b Adverbial Modifier

In some scenarios, adverbial modifiers would be present as shown in the following example. E.g., your team is highly effective. “Highly” is considered as negative in IT domain since it is used in scenarios like high priority incident, higher chance of escalation, etc., Since, “effective” is positive and “highly” is an adverbial modifier therefore, the score is changed to positive.

Furthermore, linkage between words in a sentence is established using a dependency parser tree, which is available in the spacy NLP library. A parse tree or derivation tree is an ordered, rooted tree that represents the syntactic structure of a string according to some context-free grammar. Either parse trees are constructed based on the constituency relation of constituency grammars (phrase structure grammars) or the dependency relation of dependency grammars. After the application of these heuristics, the score of a work log is the sum of scores of individual tokens. Work logs are classified into 3 categories according to rules:

1. Score  $> 0$  : Positive Sentiment
2. Score  $< 0$  : Negative Sentiment
3. Score = 0 : Neutral Sentiment

### 5 RESULTS AND DISCUSSION

In order for us to view and take actions upon those negative sentiment incidents – we built an application using AngularJS and Java. The sentiment analysis scripts are written in Python. All the screenshots used for explanation of results below are from the same application.

Fig. 1. shows the trend as to how the incident work logs went i.e. whether the trend was positive, negative or neutral. Figure. 2. shows the individual work logs and the corresponding score for the same. As can be seen in the below example as well – the incident started with positive sentiment with the customer asking for some inputs. In the next work log, we see that the user shows a negative sentiment when he/she questions as to why the case is resolved since his/her issue still persists. Furthermore, we observe that there was no response from the agent’s side due to which the sentiment goes extremely negative as the customer has used words like immediately, reopened, urgent & since no one responded to the same, we get a highly negative sentiment score. This dashboard helps in understanding the sentiment scores for different work logs and the trend as to how an incident is progressing.



Fig. 1. Trend of sentiments for incident worklogs

Actual Text	Case Status	Sentiment Score	Sentiment Type	Negative Words
Thanks for the prompt response .Please do let me know at the earliest. ...less	Cancelled	0.49	Positive	
I don't know why this case is marked as resolved. My issue still persists. Can someone look into this at the earliest and respond asap! Regards, Aarush Gupta ...less	Cancelled	-0.52	Negative	asap
There has been no update on this and as I mentioned this is urgent. I want this case to be re-opened immediately! I want my issue fixed! ...less	Cancelled	-1.56	Negative	Immediately,reopened,urgent
Hi Amrutha, Thank you for the resolution. You can mark this case as resolved. Regards, Aarush Gupta ...less	Cancelled	0	Neutral	

Fig. 2. Individual incident worklogs with their respective scores and sentiment type

We will now look at accuracy levels and some other details observed via the various tables shown below. A random sample of 100 worklogs have been considered to measure the sentiment score. Table 1 shows the results for the original VADER sentiment approach that we undertook for the IT domain. As can be seen, the VADER approach was insufficient to measure the incident worklogs sentiments. Thus, we came up with the enhanced approach to VADER sentiment and the results of the same can be seen in Table 2

Metrics	Positive	Negative	Neutral
Number of worklogs	16	35	49
Number of worklogs with sentiments captured correctly	-	3	44
Recall	0%	8.6%	90%

TABLE 1

As can be observed in Table 2, we see a much more efficient approach for measuring sentiments in IT domain incident worklogs. We have a high degree of accuracy in identifying positive and negative sentiment worklogs.

Metrics	Positive	Negative	Neutral
Number of worklogs	16	35	49
Number of worklogs with sentiments captured correctly	15	30	28
Recall	94%	85%	57%

TABLE 2

Upon usage of the application built by us in different portfolios of our organization, there was a decrease in negative sentiment cases in just one quar-

ter. Table 5.3 shows the results of the same. As can be observed from the table 5.3. the second column consists of incident count and we see a decrease in the overall incidents due to the sentiment analysis done over previous incidents.

TABLE 3

Incident Type	Q3FY2017	% of Total	Q4FY2017	% of Total
Negative Incidents	1727	7.6%	1,394	6.9%
Neutral or Positive Incidents	21,112	92.4%	18,921	93.1%
Total Incidents	22,839	100%	20,315	100%

## 6 FUTURE ENHANCEMENTS

We have to refine the domain dictionary creation process to capture more relevant words from the IT domain. Also even though the accuracy of identification of positive and negative sentiment worklogs are higher, the neutral worklog identification still requires fine tuning. We are also implementing a Convolutional Neural Network model to replace the sentiment analysis logic and to compare the performance. We will also compare the performances obtained using word2vec for embeddings with embeddings we learnt from scratch

## 7 CONCLUSION

We see a huge improvement in the performance of the sentiment analyzer in our enhanced approach over that of the conventional VADER sentiment analysis technique. This improvement can be attributed primarily to the custom created domain dictionary as well as the application of heuristics. We are implementing a Convolutional Neural Network model to further improve this performance, especially for neutral sentiment worklogs.

## REFERENCES

- [1] C.J. Hutto, Eric Gilbert, VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text
- [2] Oaindriila Das, Rakesh Chandra Balabantaray, Sentiment Analysis of Movie Reviews using POS tags and Term Frequencies
- [3] Deepak Singh Tomar ,Pankaj Sharma , A Text Polarity Analysis Using Sentiwordnet Based an Algorithm
- [4] Wei Yen Chong, Bhawani Selvaratnam, Lay-ki Soon,Natural Language Processing for Sentiment Analysis: An Exploratory Analysis on Tweets
- [5] Monisha Kanakaraj, Ram Mohana Reddy Guddeti, NLP based sentiment analysis on twitter data using ensemble classifiers
- [6] Deepak Kumar Yadav, Sampada Vishwas Massey, Sentiment Analysis Based on Data Mining and Natural Language Processing