An Analysis of Model and Feature-focused Approaches of Sentiment Classification

Kanica Ahuja
UIN: 675453362
Department of Computer Science
University of Illinois at Chicago
kahuja4@uic.edu

Kushagradhi Bhowmik
UIN: 665435416
Department of Computer Science
University of Illinois at Chicago
kbhowm2@uic.edu

Md Atiqul Islam
UIN: 673402596
Department of ECE
University of Illinois at Chicago
mislam23@uic.edu

Mohit Kumar Paritosh Ghia
UIN: 667342371
Department of Computer Science
University of Illinois at Chicago
mghia2@uic.edu

Nikhita Naramsetti
UIN: 662265273
Department of Computer Science
University of Illinois at Chicago
snaram2@uic.edu

Shibin Mathew
UIN: 656794467
Department of Computer Science
University of Illinois at Chicago
smathe36@uic.edu

Abstract

We explore contemporary context-based approach for sentiment analysis along with traditional machine learning techniques- Naive Bayes, Support Vector Machine (SVM), and Random Forest. The context-based approach identifies key terms indicative of the existence of sentiment, which are used to generate language-based models to extract features for supervised learning. Empirical evaluation of different techniques is performed over multiple real-world domains and comparison of their performance is depicted using several evaluation measures.

1 Introduction

Sentiment analysis refers to the inference of people's views, positions and attitudes in their written or spoken texts. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise as people now can, and do, actively use information technologies to seek out and understand the opinions of others. The attitude may be a judgment or evaluation, effective state (that is to say, the emotional state of the author), or the intended emotional communication (the emotional effect intended by the author). Sentiment analysis can be done at the document level, sentence level, or even sub-sentence level. A large body of work exists on the analysis of latent sentiment in social media platforms such as Twitter. The goal of these studies is to extract timely and relevant information as well as to gauge widespread opinions and sentiment [1,2,4-10].

Studies proposing learning-based algorithms can be divided into two groups: the features focused algorithms propose new features, whereas the models focused algorithms propose new classification models. The former group concentrates on creating new features that enable new perspectives of the analyzed data. These features are then fed to commonly used classifiers (such as SVM [3] and Naïve

Bayes [11]) in order to classify the sentiment of the text. The latter group is focused on proposing new algorithms and classification models.

The features-focused approaches presented in the literature are diverse, proposing both text-based features and the integration of information from multiple sources. Existing article proposed using term-frequency (TF), additional information about specific phrases, lexicon and learning-based approaches[2],[9],[10].

In this project, We presented Context-based Sentiment analysis (ConSent)[5] and other baseline methods for sentiment analysis. We tested our approach on benchmark dataset (IMDB movie reviews) and provided a comparison between commonly used classifier and the presented technique.

1.1 Approach

In this project, we focus on evaluating the emotion(positive or negative) being expressed via written text at a document level. We used Context-based Sentiment analysis (ConSent)[5], an approach for sentiment analysis effective both for regular texts (those that adhere to the rules of grammar and use existing words) and texts with a high degree of noise.

Our approach consists of two phases:

1. Learning Phase-
    - Apply techniques from the field of information retrieval to detect key terms in the text and analyze the context in which they appear.
    - Use the detected terms to generate features for supervised learning.
2. Detection Phase-
    - Use key terms and their related context terms identified in the learning phase to generate features from the test document.
    - Use the features to classify the document as either positive or negative sentiment.

1.1.1 ConSent learning phase

Key terms are detected by language modelling. We use a unigram model with terms consisting of single words. Each term is assigned a value which is the probability that it will be selected on a random sampling of words from that document. The key term identification process is performed iteratively and separately for positive and negative sentiment. If ratio of probability of a word occuring in positive context to that of it occurring in negative context is higher than a threshold value, we include that in the list of key terms for the positive sentiment; we do likewise for the negative sentiment with the above ratio inverted. Dirichlet smoothing is used when calculating term scores for indicativeness to account for possible biases arising due to data sparseness.

Since same key terms may occur with similar regularity in both positive and negative context, we consider the context they occur in by looking at terms in a window(of size six) centered around the key term. We generate language models for all such context terms from positive and negative documents. Then for each candidate for context terms, we calculate a score which is the difference between its probability for occurring in positive and negative sentiment documents. The difference is used instead of ratio as in the case of key terms to eliminate candidates which have the same ratio in both contexts, but actually appear less frequently. Candidates whose scores are above a threshold are selected as context terms, again both for positive and negative context separately.

The key and context terms, their calculated scores and positions are then used to generate a set of 41 features. Some features related to key terms are number of key terms in the document, the number of key terms without context terms and their percentage of the total number of key terms in the document, the maximum, average and standard deviation of the number of times each key term has appeared in the analyzed text, and so on. Some features related to context terms are the maximal, average and standard deviation of the number of the context terms per key term in the analyzed document, maximal, average and standard deviation of the average scores of the context terms found in the document, per key term, among others.

In the final step of the learning phase, we use the generated features to train a supervised learner, a Rotation Forest[2], which aims to build diverse and accurate classifiers. It consists of applying

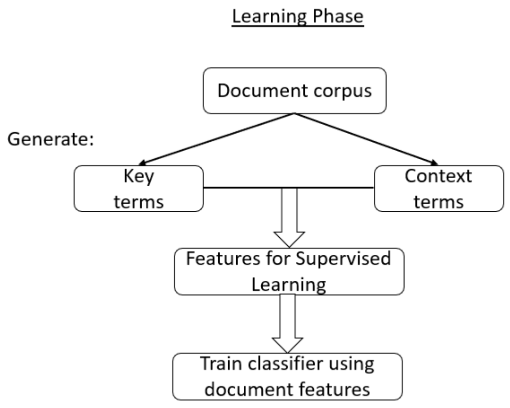
Learning Phase
Document corpus
Generate:
Key terms
Context terms
Features for Supervised Learning
Train classifier using document features

Figure 1: Learning Phase

feature extraction to subsets of features (using Principal Component Analysis) and reconstructing a full feature set for each decision tree classifier in the ensemble.

1.2 Advantages

Consent has several traits that make it very suitable for sentiment analysis in general and noisy text in particular:

- Considers the context of terms, which is important when the text is noisy.
- Does not rely on grammatical structures, which may not be reliable in noisy text.
- Enables an easy integration of information from additional sources, such as the metadata of the analyzed text (e.g., the length of the text, the time of creation, the use of emoticons, etc.)

Finally, We tested our approach on benchmark dataset (IMDB movie reviews) and provided a comparison between commonly used classifier and the presented technique.

2 Dataset

We tested on the following benchmark datasets.

- Large Movie Review Dataset v1.0 [6]
  - 50,000 reviews split evenly into 25k training and 25k test sets.
  - The overall distribution of labels is balanced (25k pos and 25k neg).
  - Neutral reviews are not included - Negative review are movies with rating <=4, Positive review are movies with rating >=7 (on 10-point scale).
  - No more than 30 reviews for any given movie because reviews for the same movie tend to have correlated ratings.

3 Context-based Sentiment Analysis

Context-based Sentiment Analysis (conSent) consists of two phases: we first apply techniques from the field of information retrieval to detect key terms in the text and analyze the context in which they appear. Then, we use the detected terms to generate features for supervised learning.

3.1 The Learning Phase

The learning process consists of three steps:

1. Identify key terms – terms indicator of the sentiment.

2. Identify context term for each key term – to model text in key term vicinity
3. Use these terms to create features

3.1.1 Key Term Identification Process

The key term identification process is carried out iteratively and separately for the positive and negative sentiment. In the interest of clarity, the key terms identification process described below is only for positive sentiment even though the process is also repeated for negative sentiment. The key terms identification process is as follows:

1. Generate a language model for the set of positive-sentiment documents (T_{pos}). Denote this language model as lm_{pos}.
2. Generate a language model for the set of negative-sentiment documents (T_{neg}). Denote this language model as lm_{neg}.
3. For each term t in lm_{pos} compute $score(t)$, which is the ratio of its frequency in positive-sentiment and negative-sentiment documents.
$$score(t) = \frac{lm_{pos}(t)}{lm_{neg}(t)} \tag{1}$$
4. If $score(t)$ exceeds a predefined threshold, t is identified as a key term.

Algorithm 1 Identifying the key terms in the training set.

```
    Generate_Context_Terms($T_{pos}, T_{neg}, key\_terms\_val\_threshold$)
1:  $key\_terms\_list \leftarrow \emptyset$
2:  $lm_{pos} \leftarrow$ Generate_Language_Model($T_{pos}$)
3:  $lm_{neg} \leftarrow$ Generate_Language_Model($T_{neg}$)
4:  for each $t$ in $lm_{pos}$ do
5:      $term\_indicativeness = lm_{pos}(t)/lm_{neg}(t)$
6:      if $term\_indicativeness \geq key\_term\_val\_threshold$ then
7:          $key\_terms\_list.Add(t, term\_indicativeness)$
8:      end if
9:  end for
```

3.1.2 Context Term Identification Process

To identify context terms for each of the identified key terms, we generate a language model for each document excerpt that is associated with an identified key term in the positive and negative documents. We then compute the score of the terms that surrounds the key terms in each document excerpt in order to identify context term. The process of identifying the context terms for a single key term (denoted as t_{key}) is as follows:

1. Locate all instances of t_{key} both in T_{pos} and in T_{neg}. If a document contains several instances of the same key term, the following steps are applied for each instance separately.
2. For each instance found in T_{pos}, extract the terms located around t_{key} by using a sliding window of size X denoted as the context span. $x/2$ terms before and after the position of the key term are extracted. Every term in this text excerpt, aside from the key term itself, is considerd a possible context term.
3. We denote each of the above text excerpts as a document excerpt d'. Excerpts from positive and negative documents will be denoted as d'_{pos} and d'_{neg}, accordingly. The set of all d'_{pos} will be denoted as D'_{pos} and D'_{neg} is similarly defined.
4. We generate language models for D'_{pos} and D'_{neg} and denote them $lm_{D'_{pos}}$ and $lm_{D'_{neg}}$ accordingly.
5. Next, for each candidate context term $t_{context}$ in $lm_{D'_{pos}}$ we calculate its score using following equation
$$score(t_{context}) = lm_{D'_{pos}}(t_{context}) - lm_{D'_{neg}}(t_{context}) \tag{2}$$

6. If the score of $t_{context}$ exceeds a predefined threshold, it will be defined as a context term for the key term t_{key}.

Algorithm 2 Identifying the context terms in the training set.

```
    Generate_Context_Terms($T_{pos}, T_{neg}, key\_terms\_list$
                $context\_term\_val\_threshold, context\_span$)
 1: for each $kt$ in $key\_terms\_list$ do
 2:     $text\_sections_{pos}$ ← Generate_Text_Sections_Containing_KT($kt, T_{pos}, context\_span$)
 3:     $text\_sections_{neg}$ ← Generate_Text_Sections_Containing_KT($kt, T_{neg}, context\_span$)
 4:     $lm_{pos}$ ← Generate_Language_Model($text\_sections_{pos}$)
 5:     $lm_{neg}$ ← Generate_Language_Model($text\_sections_{neg}$)
 6:     for each ($t$ in $lm_{pos}$) do
 7:         $term\_indicativeness = lm_{pos} = lm_{neg}$
 8:         if $term\_indicativeness \geq context\_term\_val\_threshold$ then
 9:             $context\_terms\_list.Add(t, term\_indicativeness)$
10:         end if
11:     end for
12: end for
```

3.1.3 Features Generation

Following the completion of the two previous steps, we now possess a set of key terms and their context terms. We use these terms to generate features that are in turn utilized to train a classifier for the purpose of sentiment detection. Next we present and describe these features.

As the sentiment analysis is done at the document level, we need to generate our features for each document in the training set—both with positive and negative sentiment. This process, described in Algorithm 3, consists of two steps. First, we locate the key terms and their corresponding context terms in the document. Then, we generate our proposed features set based on the identified terms.

Algorithm 3 Locating key and context terms in each document in the training set and generating features.

```
    Locate Terms And Generate Features($D, key\_terms\_list, context\_terms\_list$)
 1: for each ($d \in D$) do
 2:     for each ($kt \in key\_terms\_list$) do
 3:         if ($!d.Contains(kt)$) then
 4:             Continue
 5:         end if
 6:         $kt.positions$ ← Locate_Key_Terms_Positions ($d, kt$)
 7:         $kt.context.terms$ ← Locate_Context_Terms_Positions ($d, kt.context\_terms$)
 8:     end for
 9:     Generate Features For Document (D, d, key_terms_list)
10: end for
```

3.1.4 Generating Features for the analyzed Document

Key terms-based features: This set of features is used to provide various aspects of the detected key terms: their presumed indicative value, their proximity to other key terms in the document, the number of times each of them appears in the text, etc.

- KT_Num – the number of key terms in the document.
- KTS_max, KT_Savg & KTS_stdev – the maximal, average and standard deviation of the scores of the key terms in the document.
- KTNCT_cnt & KTNCT_perc – the number of key terms without context terms, and their percentage of the total number of key terms in the document.
- KTFmax, KTFavg & KTFstdev – the max, average and standard deviation of the number of times each key term has appeared in the analyzed text.

- KTIW10, KTIW20, KTIW30 – the maximal number of key terms that can be found in the document within a sliding window of 10,20,30 terms.
- KTLMavg & KTLMstdev – the average and standard deviation of the language model values of the key terms in the document.

Context terms-base features:

- CTNmax, CTNavg & CTNstdev – the maximal, average and standard deviation of the number of the context terms per key term in the analyzed document.
- CTPmax, CTPavg & CTPstdev– for each key term detected in the text, we compute the percentage of the context terms that were found in the analyzed text out of all those that were defined for it in earlier stages of the learning phase.
- CTSmax CTSavg & CTSstdev– the maximal, average and standard deviation of the average scores of the context terms found in the document, per key term.
- CTSRmax, CTSRavg & CTSRstdev – for each key term, we calculate the average score of its context terms that appear in the text.
- CTFmax, CTFavg & CTFstdev – the max, average and standard deviation of the times each context term has appeared in the document.
- CKTD & CKTT – the number of key terms which shared at least one context term in the analyzed document and in the entire training set, respectively.
- CTLMavg & CTLMstdev – the average and standard deviation of the language model values of the context terms detected in the document.
- SCTDmax, SCTDmin, SCTDavg & SCTDstdev – for every pair of key terms, we count the number of their shared context terms detected in the analyzed document.

3.2 The Detection Phase

The purpose of the detection phase is to utilize a learning-based classifier (in our experiments the RotationForest [8] algorithm) to assign a score to new documents. This score indicates the classifier's level of certainty that the text of document contains the type of sentiment that we are interested in detecting. In detection process takes place as follows:

1. Text of the newly analyzed document is scanned for the key terms.
2. For each detected key term, we scan the text around it in search of its context terms.
3. Finally, using the key and context term we generate the features presented and use a classifier to produce a score for the document.

Algorithm 4 Locating key and context terms in a document during the detection phase and calculating its score.

```
    Calculate Document Score($d, key\_terms\_list, context\_terms\_list$)
 1: for each ($kt \in key\_terms\_list$) do
 2:     if ($!d.Contains(kt)$) then
 3:         Continue
 4:     end if
 5:     $kt.positions \leftarrow$ Locate_Key_Terms_Positions ($d, kt$)
 6:     $kt.context.terms \leftarrow$ Locate_Context_Terms_Positions ($d, kt.context\_terms$)
 7: end for
 8: $document\_features \leftarrow$ Generate Features For Document (D, d, key_terms_list)
 9: $document\_score \leftarrow$ Classify Document ($Classifier, document\_features$)
10: return $document\_score$
```

4 Evaluation

We compare the performance of our method with three well-known benchmarks: the first two – Support Vector Machines (SVM) [3] and Naïve Bayes [11] – are both applied on unigram term

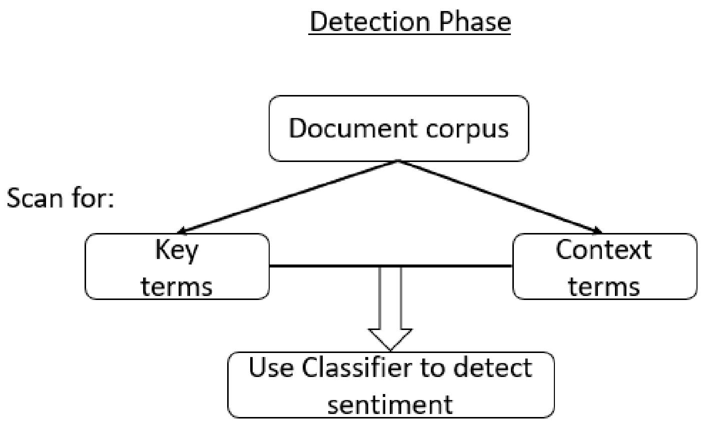

Figure 2: Detection Phase

vectors extracted from the analyzed text. These algorithms are widely used for text classification and are reported to perform well in sentiment analysis tasks [5]. The third benchmark is the Random Forest Classifier.

We use two measures in order to evaluate the performance of the algorithms:

1. Accuracy – this measure presents the overall percentage of correctly classified instances, regardless of their type. This measure is the one used to evaluate the performance of sentiment analysis.
2. Area under the curve (AUC) – this measure calculates the area under the Receiver Operating Characteristic (ROC) curve.

Although accuracy is the most commonly-used measure for sentiment analysis tasks, we maintain that there is good reason to consider the other measure as well. The Receiver Operating Characteristic (ROC) curve is a graph produced by plotting the true-positive rate versus the false-positives rate over all possible acceptance thresholds. The area under this curve – the AUC – evaluates both classes simultaneously and at various decision points. Therefore, the classifier's performance is not overwhelmed by the majority class [5].

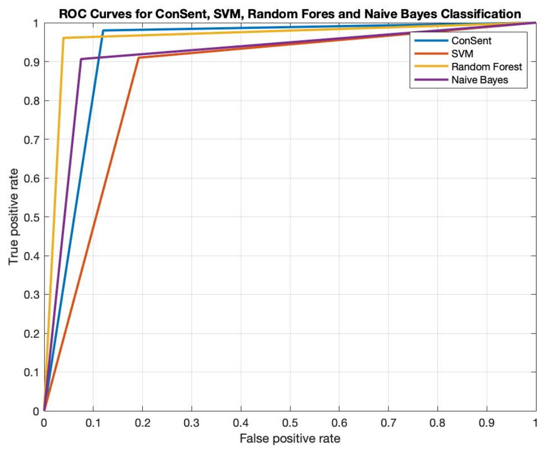

Figure 3: Receiver Operating Characteristic (ROC) curve using IMDB Dataset

Techniques Used	**Accuracy**
ConSent	93.3%
Naive Bayes	91%
SVM	87%
Random Forest	95%

5 Result and Comparison

We used the movie review dataset to compare the performance of the ConSent approach to those of the three baselines. The result for the accuracy is presented in Table 1. ConSent approach outperforms the SVM and Naive Bayes Methods. Random Forest classifier achieves the highest accuracy. The ROC curve for the approaches used is illustrated in Figure 3. It shows that ConSent method clearly outperforms other baseline methods.

6 Summary and Future Work

In this project, we studied different approaches for sentiment analysis. We investigate context based approach (ConSent) built on techniques from the field of information retrieval in order to to identify key terms which are indicative of sentiment and the context in whic they appear. Our evaluation demonstrates that the ConSent approach outperforms other baseline methods (SVM, Naive Bayes). The strength of our approach stems both from its focus on key terms and its heavy reliance on context.

In future work we can consider using di-gram or tri-gram models for the language modelling step for key and context terms. More features can be extracted and used from the statistics involving the key and context terms. We can also include some metadata features about the documents themselves, such as overall length of the document under consideration.

References

[1] Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik Van Der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. Sentiment analysis in the news. *arXiv preprint arXiv:1309.6202*, 2013.

[2] Erik Boiy and Marie-Francine Moens. A machine learning approach to sentiment analysis in multilingual web texts. *Information retrieval*, 12(5):526–558, 2009.

[3] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing multiple parameters for support vector machines. *Machine learning*, 46(1-3):131–159, 2002.

[4] Nadia FF Da Silva, Eduardo R Hruschka, and Estevam R Hruschka Jr. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66:170–179, 2014.

[5] Gilad Katz, Nir Ofek, and Bracha Shapira. Consent: Context-based sentiment analysis. *Knowledge-Based Systems*, 84:162–178, 2015.

[6] Andrew L Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies-volume 1*, pages 142–150. Association for Computational Linguistics, 2011.

[7] Bo Pang, Lillian Lee, et al. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1–2):1–135, 2008.

[8] Juan José Rodriguez, Ludmila I Kuncheva, and Carlos J Alonso. Rotation forest: A new classifier ensemble method. *IEEE transactions on pattern analysis and machine intelligence*, 28(10):1619–1630, 2006.

[9] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics, 2005.

[10] Qiang Ye, Ziqiong Zhang, and Rob Law. Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert systems with applications*, 36(3):6527–6535, 2009.

[11] Harry Zhang. The optimality of naive bayes. *AA*, 1(2):3, 2004.