IJCSI International Journal of Computer Science Issues, Volume 13, Issue 5, September 2016
ISSN (Print): 1694-0814 | ISSN (Online): 1694-0784
www.IJCSI.org https://doi.org/10.20943/01201605.120124

Unigram Polarity Estimation of Movie Reviews using Maximum Likelihood

Rounak Dhaneriya[1], Manish Ahirwar[2] and Dr. Mahesh Motwani[3]

[1] Department of Computer Science & Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India

[2] Department of Computer Science & Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India

[3] Department of Computer Science & Engineering, Rajiv Gandhi Proudyogiki Vishwavidyalaya Bhopal, India

Abstract

This research work focuses on sentiment analysis, the detection of polarity and estimating the intensity of polarity of movie reviews. Internet movie database (IMDB) is the source of data named polarity dataset version 2.0 which is used in this research. There are 1000 reviews of movies for each category positive and negative. Unigram based Maximum likelihood algorithm is used which uses logarithmic likelihood ratios for estimating intensity and detection of polarity. This supervised technique is able to deal with complex sentences and detecting polarity of words. This approach uses unigram models to detect polarity and uses likelihood ratios for calculating the intensity. The results suggest that the sentiment analysis using unigram based maximum likelihood logic performs well.

Keywords: *Sentiment Analysis, Polarity Detection, Intensity Estimation, Maximum Likelihood.*

1. Introduction

Sentiment Analysis (SA) is the field of study that examines individual's sentiment, opinions, evaluations, attitudes, appraisals and feelings towards entities, for example, products, services, organizations, and their aspects [1]. Sentiment Analysis is often referred as opinion mining in many contexts. Sentiment analysis is defined as the task of analysing text computationally with the help of machine learning techniques and data mining approaches. It determines the opinion expressed by the author for particular objects or entities. Sentiment Analysis mainly uses three kinds of approaches namely Machine learning based, Lexicon based and Hybrid approach (Machine Learning & Lexicon Based approaches) for identifying opinion expressed by the user [2]. The main aim of sentiment analysis is to process and label the opinion in different categories such as positive opinion and negative opinion. Another task of sentiment analysis is to determine either the given source is subjective or objective, expressing the purely facts about the writer's opinion. These tasks were performed at different level of analysis which can be categorized as document level, sentence level and word level sentiment analysis on the source [3].

Textual representation of web pages leads to analysis on this web based text which is termed as online information retrieval. Online information retrieval includes extraction of text, splitting of text into parts, checking spellings and counting frequency of specific words. Sentiment analysis allows to transform (unstructured) textual information to (structured) machine-processable data to extract potentially meaningful information. The next sections represent survey and discusses about the sentiment analysis methods, techniques and process without focusing on specific task and review main research problem in recent articles presented in this field. This research paper focuses on basic n-Gram models especially unigram models to analyse sentiment of review text.

This kind of models is derived from an approximation of the probability of a sequence of words, which is based on a Markov property assumption. Let us consider, for instance, a unit of text w which consist of a sequence of words w1, w2,…wm. The probability of such a sequence can be decomposed, by means of the chain rule, in the following product of probabilities:

$$p(w) = p(w_1, w_2, \ldots w_m) \quad \ldots(1)$$

$$= p(w_1)p(w_2|w_1)p(w_3|w_1,w_2)\ldots p(w_m|w_1,w_2\ldots w_{m-1}) \quad \ldots(2)$$

A Markov process refers to a random process in which the probability of the next state only depends on the current state, and it is statistically independent on any previous states. In our specific context of word sequences, assuming the Markov property implies considering that the probability of a given word only depends on a fixed

number of preceding words. According to this, n-gram models are defined by approximating the conditional probabilities in (1) and (2) by conditional probabilities that only depend on the previous n-1 words in the sequence (commonly referred to as the history). In this way, n-gram models of order one (1-gram), two (2-gram), three (3-gram), and so on, can be defined as follows:

1-gram: $p(w) \approx p(w_1)p(w_2)p(w_3)\ldots p(w_m)$ …(3)

2-gram: $p(w) \approx p(w_2|w_1)p(w_3|w_2) p(w_4|w_3) \ldots p(w_m|w_{m-1})$ …(4)

3-gram: $p(w) \approx p(w_3|w_2,w_1)\ p(w_4|w_3,w_2) \ldots p(w_m|w_{m-1},w_{m-2})$ …(5)

n-gram: $p(w) \approx \Pi_i p(w_i|w_{i-1},w_{i-2}\ldots w_{i-n+1})$ …(6)

Maximum likelihood estimates can be easily computed for probabilities in (3), (4), (5) and (6) by using a training corpus. For small values of n, the probabilities in (3), (4), (5) and (6) are much easier to estimate than those in (1) and (2). Indeed, when long word histories are involved, the model tends to become unreliable as most of the histories are not actually seen in the training dataset and the corresponding n-gram probability estimates are not reliable. Also, notice that in the extreme case of the unigram (3), the resulting model is completely independent of the order of words. Such a word-order independent model is known as bag-of-words model. Different from the unigram case, in the bigram, the trigram and the general n-gram model (4), (5) and (6), word order is taken into account. In the bigram case, the probability of a given word depends on the word immediately before, in the trigram case, the probability of a word depends on the previous two words, and so on [4].

2. Literature Survey

P.D. Turney[5] introduce work on SA concentrated on recognizing extremity of reviews on automobile bank, movie and travel and B. Pang et.al[6] proposes a document level analysis on motion picture surveys from IMDB (Internet Movie database).

T. Nasukawa et.al [7] shows a way to deal with SA in which they separate sentiments for specific subjects with its corresponding polarities i.e. negative or positive from a document, regardless of classifying the entire record.

Later work handles SA at sentence level in "Mining and Summarizing Customer Reviews" M. Hu et.al[8].X. Ding et.al [9]proposed a powerful technique for recognizing semantic introductions of opinions communicated by reviewers on product features.

In Sentiment Analysis, A. Go et.al [10]presents a framework, which can extract micro blogged messages relevant to a particular topic from a blogging site, for example, Twitter and afterward analyse the messages to decide sentiments they convey and to characterize them as unbiased, positive or negative.

Current studies centre was moved from sentence level to phrase level in "Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-grams"A.Agrarwal et.al [11] and O'Connor [12] presents short-message frames in light of the ubiquity of micro blogging sites such as Twitter.

M. Trupthi et.al [13] used N-grams, NB, NLP Techniques for the task of sentiment classification on movie review dataset while M. Gamon[14] and B. Pang et.al[15] used supervised machine learning techniques for the movie reviews classification.

R. Arora and S. Srinivasa[16] used N-gram supervised classification on the user generated content while V. Singh et.al [17] used movie review data and R. prabowo et.al [18]used movie review, product review and MySpace Comments data and applied feature-based heuristic aspect-level SA using N-grams and SVM, Rule based classifier for the classification accordingly.

Z. Wang et.al [19] proposes Fuzzy inference method (FIM)with linguistic processors unsupervised classification on social media data while Y. Zhao et.al [20] proposes lexicon based SAMC algorithm for the task of classification on same data.

3. Proposed work

This work is implemented on MATLAB computing environment. We used a standard experimental polarity dataset that has been originally derived from the IMDb collection. This dataset, known as polarity dataset v2.0, contains full texts of 1,000 positive and 1,000 negative reviews on movies, and it is available from the Movie Review Data website at Cornell University (http://www.cs.cornell.edu/people/pabo/movie-review-data/).Next, we applied the pre-processing to the files and load the complete data collection into two structure arrays, data_pos and data_neg. Next, we partition each of the two category subsets into three subsets: train, test and development. We used randomly permuted indexes in the data file randpermutations.mat: where the first 800 indexes contained in randselect_pos and randselect_neg define the train set, the following 100 indexes define the test set, and the last 100 indexes define the development set.

Table 1: Basic statistics for both, positive and negative, review subsets and the overall polarity dataset v2.0 collection

	Positive reviews	Negative reviews	Full collection
Number of documents	1,000	1,000	2,000
Running words	705,134	631,749	1,336,883
Vocabulary size	30,218	28,255	39,339
Minimum document length	120	17	17
Maximum document length	2,462	1,888	2,462
Average document length	705.13	631.75	668.44

In this section we will be approaching polarity detection and intensity estimation problems from the statistical perspective. More specifically, unigram based a naïve Bayes approach, based on the likelihood ratio method. Unigram-based models have been computed for each data category by using the corresponding train subsets. The models are available in the data file named unigram_model.mat, which contains the two structures unigram_pos and unigram_neg. Each structure is composed of the three fields: vocab, prob and unk_prob, which contain the vocabulary terms, the unigram probability estimates for the vocabulary terms and the unseen event probability estimates, respectively. Logarithmic likelihood ratios can be computed for a given document or document set by using the function compute_loglirat. This function implements a unigram version of the log likelihood ratio computation procedure. The syntax of the function is as follows:

Loglirat = compute_loglirat (dataset,model1,model2) …(1)

where model1 and model2 are structures containing unigram models in the same format as described above for unigram_pos and unigram_neg, and dataset is a structure array containing one or more documents in the same format as the two structures arrays data_pos&data_neg. The text must be provided in the form of a cell array of tokens under the structure field dataset.token. The output returned by the function compute_loglirat is a numeric vector containing as many values as documents are given in the input structure dataset. For each individual document d, the function returns the result of the following operation:

Loglirat = log(p(d|model1))-log(p(d|model2)) …(2)

where the document probability p is estimated, according to the naïve Bayes criterion, as a product of unigram probabilities. A log likelihood ratio greater than zero suggests that the given document is more likely to belong to the positive category, and a log likelihood ratio less than zero suggests that it belongs to the negative category.

Algorithm: Unigram log-likelihood ratio estimates

```
1: INPUT: tst_data_pos, tst_data_neg, unigram_pos, unigram_neg
2: OUTPUT: loglirat_pos, loglirat_neg
---------------------------------------------------------------------
3: Initialise: logprob_pos← zeros(length(data),1)
4:             logprob_neg← zeros(length(data),1)
5: for all k = 1 to length(data) do
6:     for all n = 1 to length(data(k).token) do
7:         token ← data(k).token{n}
8:         idx← find(strcmp(model1.vocab,token))
9:         ifisempty(idx) then
10:        c1 ← model1.unk_prob
11:        else c1 ← model1.prob(idx) end
12:        logprob_pos(k)←logprob_pos(k)+
                          log2(c1)
13:        idx← find(strcmp(model2.vocab,token))
14:        ifisempty(idx) then
15:        c1 ← model2.unk_prob
16:        else c1 ← model2.prob(idx) end
17:        logprob_neg(k)←logprob_neg(k)+
                          log2(c1)
18:    end for
19: end for
20: loglirat←logprob_pos-logprob_neg;
21: return (loglirat)
```

Next the problem of intensity estimation, in which we are interested in assigning a certain degree of positivity or negativity to a given document. The computed log likelihood ratios already provide a means for intensity estimation as the computed values are actually distributed over a wide range of real values. Indeed, we can intuitively think about documents exhibiting larger positive log likelihood ratios as being ‘‘more’’ positive than those exhibiting smaller positive log likelihood ratios. Similarly, we can think about documents with larger negative ratios as being ‘‘more’’ negative than those with smaller negative ratios.

According to this, the computation of log likelihood ratios already addresses the problem of intensity estimation, and we derived the polarity detection from it by using the zero value as a threshold for the corresponding binary decision. Pure log likelihood ratio values do not seem to be a good choice as intensity scores. Indeed, in theory, such values can range from minus infinity to infinity, making it very difficult to appreciate how positive or negative are the values. In order to have a more appropriate range of values for estimating polarity intensity, we should use a normalization function to convert pure log likelihood ratios into a more useful polarity intensity score. Here we used a sigmoid function for conducting the normalization. More specifically, we used the following normalization formula:

$$\text{intensity\_score} = 2 / (1+e^{-\text{loglirat}}) - 1 \quad \ldots(3)$$

The normalized intensity score proposed in (3) maps polarity intensities between the negative and positive extremes into the interval from -1 to +1, respectively.

4. Results

We detected polarity and computed its intensity for sentences. In the following sentence, we illustrated the use of the log likelihood ratio as estimator of polarity intensity. Consider, for instance, the three sample sentences: the movie is bad, the actor performance is excellent and the actor performance is excellent but the movie is bad. The calculated log likelihood ratios for the above sentences are -1.9795, 2.2986 and 0.3013 and calculated intensity score for above sentences are -0.7573, 0.8175 and 0.1495. The twelve examples presented in the table have been ranked in ascending order according to their polarity intensities, from the most negative (in rank 1) to the most positive (in rank 12).

Table 2: Some samples on polarity detection and intensity estimation within the movie review domain.

	Samples	Loglirat	Polarity	Score
1.	It was as good as garbage.	–2.7479	Negative	–0.8796
2.	This actor is terrible; his performance was pathetic.	–2.8371	Negative	–0.8893
3.	The music was bad and the script was boring.	–4.3772	Negative	–0.9752
4.	This film has some problems with the plot.	–1.0615	Negative	–0.4860
5.	Not as good as the previous movie in the saga.	2.0801	Positive	0.7779
6.	Not so bad. I was expecting something worst.	–5.0530	Negative	–0.9873
7.	The plot was simple, but I enjoyed the movie anyway.	–1.0728	Negative	–0.4903
8.	Interesting film, which is full of action and excitement.	–0.3436	Negative	–0.1701
9.	A very funny and pleasantly entertaining film.	0.7424	Positive	0.3550
10.	Wonderful script and beautiful photography. A great movie!	2.3287	Positive	0.8225
11.	Excellent movie, as expected from such a great director.	1.0857	Positive	0.4951
12.	Exceptional production I will be watching again and again.	1.3771	Positive	0.5971

5. Conclusion

For polarity detection and calculating intensity of sentiments we used unigram based maximum likelihood approach. The inputs are in the form of textual data with their respective characteristics such as, vocabulary, index, tokens, text and categories etc. and the output is the detected polarities in different categories and respective sentences with their intensities scores and loglirat ratios. As seen from the loglirat ratios and scores, with the exceptions of good and normal, most of the intensity estimates for the considered words are correct. In the case of good, however, it has been assigned an intensity score close to zero instead of a more appropriate high positive value. Similarly, the word normal received and exaggeratedly high positive score rather than a more appropriate close-to-zero value, such as in the case of regular.There is need of NLP (Natural Language Processing) techniques to be applied to this review content for estimating better results. A hybrid technique with the use of NLP and maximum likelihood approach must be

made for the better estimation of intensity of polarity and analysing sentiment from the review data.

References

[1] B. Liu, Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, 2012.

[2] Z. Madhoushi, A. R. Hamdan and S. Zainudin, "Sentiment Analysis Techniques in Recent Works," in *Science and Information Conference* , London, UK, 2015.

[3] I. K and R. R. P C, "Fuzzy Logic Based Sentiment Analysis of Product Review Documents," in *IEEE, International Conference on Computational Systems and Communications (ICCSC)*, Trivandrum, 2014.

[4] R. E. Banchs, Text Mining with MATLAB, Barcelona: Springer Science+Business Media , 2013.

[5] P. D. Turney, "Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews," in *Association for Computational Linguistics (ACL)*, Philadelphia, 2002.

[6] B. Pang, L. Lee and S. Vaithyanathan, "Thumbs up? Sentiment Classiflcation using Machine Learning Techniques," in *Empirical Methods in Natural Language Processing (EMNLP)*, Philadelphia, 2002.

[7] T. Nasukawa and J. Yi, "Sentiment analysis: capturing favorability using natural language processing," in *(K-CAP) Knowledge Capture* , 2003.

[8] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," in *KDD*, 2004.

[9] X. Ding, B. Liu and P. S. Yu, "A Holistic Lexicon-Based Approach to Opinion Mining," in *WSDM, (ACM) Association for Computing Machinery*, 2008.

[10] A. Go, R. Bhayani and L. Huang, "Twitter Sentiment Classification using Distant Supervision," *CS224N Project Report, Stanford,* vol. 1, p. 12, 2009.

[11] A. Agarwal, F. Biadsy and K. R. Mckeown, "Contextual Phrase-Level Polarity Analysis using Lexical Affect Scoring and Syntactic N-grams," in *Association for Computational Linguistics (ACL)*, 2009.

[12] B. O'Connor, R. Balasubramanyan, B. R. Routledge and N. A. Smith, "From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series," in Association for the Advancement of Artificial Intelligence, 2010.

[13] M. Trupthi, S. Pabboju and G. Narasimha, "Improved Feature Extraction and Classification - Sentiment Analysis," IEEE, 2016.

[14] M. Gamon, "Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis," Microsoft Research, 2004.

[15] B. Pang and L. Lee, "A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts," in ACL, 2004.

[16] R. Arora and S. Srinivasa, "A Faceted Characterization of the Opinion Mining Landscape," 2014.

[17] V. Singh, R. Piryani, A. Uddin and P. Waila, "Sentiment Analysis of Movie Reviews: A new Feature-based Heuristic for Aspect-level Sentiment Classification," IEEE, 2013.

[18] R. Prabowo and M. Thelwall, "Sentiment Analysis: A Combined Approach," 2009.

[19] Z. WANG, V. J. C. TONG and D. CHAN, "Issues of social data analytics with a new method for sentiment analysis of social media data," IEEE, 2014.

[20] Y. Zhao, K. Niu, Z. He, J. Lin and X. Wang, "Text Sentiment Analysis Algorithm Optimization & Platform Development in Social Network," IEEE, 2013.

[21] D. Virmani, S. Taneja and P. Bhatia, "Aspect Level Sentiment Analysis to Distil Scrupulous Opinionated Result," IEEE, 2015.

[22] A. Salinca, "Business reviews classification using sentiment analysis," IEEE, 2015.