# Effective Automation in Aspect Based Review System

**M. Venu Gopalachari**

Dept of CSE, Chaitanya Bharathi Institute of Technology,bHyderabad, India
Venugopal.m07@gmail.com

**Abstract**

E-Commerce is the field commanding the global market by highly reaching the users in providing quality of services. Text-based reviews found online have become a common way to evaluate options when making a decision. The reviews having multiple sentiments on various things of interest are common in nature and cannot be aggregated. In order to determine actual sentiments on various things called aspects evolved and useful in the field of e-commerce. In this paper an aspect based review system is proposed to identify and evaluate sentiments about aspects automatically. Experiments shown significant improvement in performance when compared with existing methodologies.

**Keywords:** Natural language processing, Aspects, sentiment analysis, e-commerce,

## 1. Introduction

Text-based reviews found online have become a common way to evaluate options when making a decision. These reviews span subjects from a variety of topics - products available for purchase online, downloadable applications, movie and music releases, restaurants, hotels, and more. Oftentimes, these reviews are associated with an overall numeric rating (typically on a 5-point or 10-point scale), which can be aggregated to form an average rating for a given subject. However, these ratings oftentimes hide the details present in the text of the reviews. For example, by examining a set of laptop reviews with an average rating of 3.0 out of 5.0, one might find that the screen of the laptop is mostly referred to positively, but the keyboard is mostly referred to negatively. This nuance is not reflected with an overall 5-point numeric rating, despite the fact that users of ten times have preferences that require a more detailed view of the subject.

In order to more accurately reflect how reviewers feel about different aspects of a subject, it is desirable to develop a system to rate the major features of a subject separately, providing more meaningful information to those who may have specific preferences. A shopper looking to purchase a laptop, for example, may desire a high screen quality while not caring much about the processing power. This shopper would benefit from finding a laptop with a highly-rated score for the aspect "Screen" and may not mind if the laptop's overall score is dragged down by a lower rating for the aspect "Processing Power". It's possible that websites aiming to have a more comprehensive set of ratings could force users to rate specific qualities on a numeric scale, rather than just the overall product. However, this requires more effort on the end user, and ignores the vast amount of text-based review data that already exists.

One way such a system can be developed using existing product reviews is to utilize sentiment analysis (also known as opinion mining). Sentiment analysis attempts to derive measures of subjectivity from written text, typically labeling text using either the labels "subjective" and "objective" (ignoring polarity of subjective text), or the labels "positive", "negative", and "neutral" (where "positive" and "negative" are opposite categories of subjectivity, and "neutral" is equivalent to "objective"). Text-based reviews are an important

source of data for sentiment analysis because they consist primarily of subjective opinions, making them particularly useful for building models with the ability to determine sentiment polarity.

However, rather than attempting to determine the sentiment of the review as a whole, the sentiment of particular attributes of the product would be measured. If a particular attribute is found to be associated with positive or negative polarity for most instances within a set of reviews, then it is given a high or low rating, respectively, for that particular attribute. These attributes (or aspects) can be found through aspect identification - determining what words and phrases (terms) refer to specific aspects of the subject. For example, in the sentence "The battery life is quite strong and lasts all day long," the phrase "battery life" is an aspect term of the subject.

Once these aspect terms have been identified, sentiment analysis can be used to determine the sentiment polarity of each aspect term. Specifically, aspect-based sentiment analysis attempts to determine the sentiment of each aspect term. Accurately determining the polarity of aspect terms is more challenging than the typical sentiment analysis task. Sentiment analysis relies heavily on sentiment lexicons that classify adjectives based on their sentiment polarity, but an adjective that has a positive sentiment when used to describe one aspect may have a negative or neutral sentiment when used to describe another aspect. For example, \long" tends to have a positive sentiment when used to refer to \battery life" in a laptop, but a negative sentiment when used to refer to \wait times" at a restaurant. Another significant issue is when multiple aspect terms are mentioned within the same sentence. If one aspect has a positive sentiment and another has a negative sentiment, determining these sentiments accurately requires understanding which portions of the sentence apply to a given aspect term.

## 2. Background

Natural Language Processing (NLP) is a field of study within computer science and artificial intelligence that focuses on analyzing and deriving meaningful information from human (natural) language. Natural language processing developed as a result of interest in machine translation (MT), the problem of translating sentences automatically from one language to another, in the 1950s. Research was severely limited due to the relatively undeveloped state of computers at the time. Initial research started as dictionary-based, with attempts to translate sentences word-for-word, but issues with determining the correct syntax (the arrangement of words) and semantics (the meaning of words) in translation quickly showed the limitations of such an approach. Despite technological limitations, research of this time period was able to effectively identify the importance of developing an explicit structure and definition for language that could allow methods to be generalized and implemented with computers [1]. The low quality of the methods developed, however, led a committee commissioned by the United States government called ALPAC (Automatic Language Processing Advisory Committee) to express doubts in the merit of continued MT research in a report in 1966 [2].The committee suggested that significant improvements in computational linguistics was needed before MT could be effectively tackled, leading to a significant shift away from MT in the late 1960s. This shift allowed other problems within NLP to be explored, eventually leading to the broad range of problems studied within the field today.

The massive amount of data and processing power that are accessible today has opened the door to new heights in the world of natural language processing. Modern NLP research examines problems such as converting speech to text [3], answering text-based questions [4],

automatically summarizing large documents, automatic spell-checking, determining grammatical relationships between words, and much more. NLP has been utilized in a large variety of business contexts as well. Lawyers use NLP software to analyze large sets of legal documents to find meaningful information. Spam filters utilize NLP to find patterns within email text that indicate a high likelihood of being spam, and Google uses NLP in their language-translation software. Various social media sites utilize natural language processing so that advertisements can be customized to the interests of each user.

In this paper, we utilize some commonly-used software for natural language processing. In particular, we make extensive use of the Natural Language Toolkit (NLTK)[5] and Stanford's Core NLP toolkit [6]. There are different types text features to be concerned with as part of text analysis.

### 2.1 Token-level Features

We break each sentence down into tokens consisting of words and punctuation using the Penn Treebank tokenizer within NLTK [7.]. This tokenizer splits contractions and stores punctuation as separate tokens. Some features can be extracted from individual tokens without the need for information from the rest of the sentence or corpus. We store the original token text, as well as a lowercase version of the token. Several binary features are stored - whether or not the token is punctuation, whether or not it is in "title case" (the first letter of the token is capitalized, and the following letters are all lowercase), and whether the token is a digit. We use a popular word stemmer, Porter Stemmer, to store the stem of a given word, removing all prefixes and suffixes from the token [8].

### 2.2 Sentence-level Features

Some features require sentence-level context. The index of each token within the sentence is stored, with 0 being the first token of the sentence. A part-of-speech (POS) tagger using the Penn Treebank tag set is used to tag the part-of-speech for each token in a sentence [9]. The full POS tag and the first 2 characters of the POS tag are stored as separate features, as the first two characters are indicative of a broader category that the following characters are part of (for example, \NN", \NNP". \NNS", and \NNPS" are all tags to describe nouns). Each token also stores information about the previous and next tokens in the sentence - the text, lowercase text, stem, and both POS tag features of the previous and next tokens, storing a default value if the previous or next token doesn't exist.

### 2.3 Review-level Features

Oftentimes, text-based reviews are associated with an overall numeric rating. Our datasets do not have contain numerical rating information, but utilizing these review ratings in an aspect-based sentiment analysis model may yield positive results, and is worth future consideration when designing annotated datasets from online reviews.

### 2.4 Other Possible Features

Many other features are commonly used for natural language processing purposes. Word-Net is a lexical database designed to store words based on their word sense (the meaning of the word) rather than the word itself [10]. It contains over 155,000 words and 117,000synonym sets (sets of words with the same meaning), with over 206,000 word-sense pairs in total [11]. Several other semantic relations are stored as well, such as antonyms. However, Word Net has been found to not significantly impact the performance of text classification models [12], and the limited tests we performed showed little benefit. Despite this, usage of Word Net in other

models for aspect identification and aspect-based sentiment analysis may still be worth exploring.Word2Vec is a deep learning algorithm that takes sentences as inputs and out-puts a vectorization of each distinct word within the training data. This can be used to determine the similarity of one word from another word. Training Word2Vec on larger datasets available, such as the full English Wikipedia, has resulted in positive results in other aspect identification models [13].

## 3. Identifying Aspects in text

In some texts, particularly text-based reviews, there is an overall subject being discussed throughout the text. Aspect identification (or aspect term extraction) is the process of identifying what words and phrases (terms) refer to specific aspects of a subject in these texts. Aspect identification typically refers to extracting aspect terms explicitly mentioned within the sentence, rather than implied terms. For example, the sentence "The restaurant was quite expensive" does not explicitly mention price, but "expensive" is an adjective referring to the price of the food, an implicit aspect within the sentence. We consider only explicit aspect terms in this paper. An ideal system would not rely heavily on the domain of the training data, as otherwise a new set of training data would be required for each new domain examined. Identifying aspect terms requires human identifiers to manually record these aspect terms and their sentiment, and requires a consistent approach so that these human identifiers mostly agree with each other. When each set of training data requires potentially hundreds of reviews (thousands of sentences), this task becomes infeasible to complete for the many domains available for text-based reviews.

One of the most significant challenges in aspect identification is balancing accuracy with robustness. The most accurate models will likely require more detailed training data - accurate sentence-level datasets identifying aspect terms and their respective polarities (positive, negative, or neutral). But the most domain-neutral models will rely on more general features and potentially unsupervised methods. Thus, we examine both supervised and unsupervised approaches, and test across domains to see how applicable each supervised method is to training data from a different domain.

### 3.1 Sequential Labeling: Conditional Random Fields

Aspect term extraction can be modeled as a sequence labeling problem, where each sentence is examined as a sequence of tokens, taking the context of an individual token into account. This framework is used for problems such as part-of-speech tagging, named entity recognition, and shallow parsing [14]. We describe and implement a common sequence labeling model called a Conditional Random Field (CRF), a generalization of another model called a Hidden Markov Model. These are sequential labeling models based on generalizations of the single-label models described with the naive Bayes classifier and Maximum Entropy models. The goal of a CRF is to determine the conditional distribution of potential labels (in our case, using the IOB2 tagging format) given the output (each token's text). Using the framework for Maximum Entropy models and CRFs, feature functions can be defined that can allow a vector of output features to be associated with each word in a sentence.

#### 3.1.1 Labeling Method

We use the IOB2 tagging format, where each token is associated with one of three labels - inside an aspect term ("I"), outside an aspect term ("O"), or the beginning of an aspect term ("B"). All aspect terms start with a "B", so only multi-token aspect terms utilize the "O" label.

### 3.1.2 Naive Bayes and Maximum Entropy Models

The naive Bayes classifier is used to predict a class label y given a feature vector x. It is based on the assumption of conditional independence of the individual features given the class label. The model attempts to maximize the joint probability p(x; y) of the features and the class label, which due to their conditional independence can be described as follows:

$$p(X, y) = p(y)\prod_{i=1}^{n} p(x_i \mid y). \qquad 3.1$$

The Maximum Entropy classifier (also known as multinomial logistic regression) makes the assumption that log(p(y j x)) can be represented as a linear combination of the features and a constant. This is useful in that the features are not assumed to be independent, and so the relationships among the output features are considered. The Maximum Entropy classifier models the conditional probability p(y j x) as follows:

$$p(y \mid x) = \frac{1}{z}\exp(\beta_y x + \beta_{y,0}). \qquad 3.2$$

$z = \sum_y \exp(\beta_y x + \beta_{y,0})$ is a normalization constant which adjusts to ensure valid probabilities. The parameters $\beta_y$ and $\beta_{y,0}$ can be chosen based on the training data using the expectation-maximization (EM) algorithm [15].

Naive Bayes is a generative model, meaning that the model estimates the joint probability distribution of the state and the feature vector and uses this learned distribution to predict the likelihood of a feature vector x being assigned a class label y. Maximum Entropy models, on the other hand, are discriminative - they learn the conditional probability $p(y \mid x)$ of being in a state x given an output y. This is important because unlike generative models, the probability distribution of outputs $p(x)$ does not need to be learned. In the case of natural language processing where the observed out-puts are words, there are almost certainly words that don't exist in the training corpus that may occur when using the model, meaning $p(x)$ cannot be accurately estimated without training data that contains every possible word - an unfeasible task.

Because these classifiers only predict a single class label for a set of features, they cannot model the relationships among the hidden states. Graphical models such as Hidden Markov Models and CRFs, on the other hand, are able to account for the dependencies between the nodes' labels.

A feature function and corresponding parameter can be defined for any function of the current features, the current label, and the previous label. The general model is described below:

$$p(y \mid x) = \frac{\exp\left[\sum_{i=1}^{n}\sum_{q=1}^{F}\lambda_q f_q(y_i, y_{i-1}, x_i)\right]}{Z(x)}. \qquad 3.3$$

Where $Z(x) = \sum_y, \exp\left[\sum_{i=1}^{n}\sum_{q=1}^{F}\lambda_q f_q(y'_i, y'_{i-1}, x_i)\right]$ is the normalization constant, computed by summing the feature functions multiplied by their weights over the possible label combinations. The number of possible label combinations becomes very large, but it will be shown that this problem can be averted during CRF training.

## 4. Results and Discussions

An important consideration is the method with which ATE systems are evaluated. One key question is whether to apply these methods to distinct aspect terms or to each occurrence of an aspect term. If we evaluate based on distinct aspect terms, then we take the set of predicted distinct aspect terms and compare them to the list of actual distinct aspect terms. However, aspect terms with higher frequency are more valuable, given that our eventual goal is to determine polarity scores for a few most common terms/categories. A model that is able to accurately predict high-frequency aspect terms, but is less effective at predicting low-frequency aspect terms, is more valuable than a model that is better at predicting low-frequency terms than high-frequency terms.

On the other hand, evaluation based on instances of each aspect term can lead to overconfidence in models that can identify some of the most common terms with accuracy, but cannot accurately identify most other terms. Aspect terms with the highest frequencies in the dataset aren't always more important to accurately identify than aspect terms with lower frequencies. An individual aspect term may be more frequent than other aspect terms simply because it has few or no synonyms (for example, "Microsoft Office" has no synonyms, while "price" has several different words representing the same concept).

Thus, we evaluate the methods described in the previous sections with respect to both distinct aspect terms and instances of each aspect term. We use 70% of the data available in each domain for training and 30% for testing. As a review, three of the most common methods of evaluating models are precision, recall, and F-measure. Precision describes the fraction of predicted aspect terms that actually exist in the dataset. Recall is the fraction of true aspect terms that are predicted by the model. F-measure is the harmonic mean of precision and recall.

CRFsuite implements several algorithms to solve for the CRF parameters. Two of the most common optimization algorithms for solving CRFs are provided: L-BFGS and stochastic gradient descent. L-BFGS is a common quasi-Newton method that avoids storing a full approximated Hessian, making is useful for problems such as CRFs where there are often a large number of parameters to be found [18.]. Stochastic gradient descent (SGD) is an extension of gradient descent that moves in the direction of a random data point at each iteration. In the CRFsuite implementation, SGD is performed with 2 regularization to prevent over fitting. Both of these algorithms have been shown to be successful when utilized to solve conditional random fields [19]

Table 3.1: The results for CRFs using distinct aspect terms.

| Algorithm | Dataset | Precision | Recall | E-measure |
|---|---|---|---|---|
| L-BFGS | Restaurants | 0.7003 | 0.5224 | 0.5984 |
| SGD | Restaurants | 0.6095 | 0.4187 | 0.4964 |
| AP | Restaurants | 0.6701 | 0.4004 | 0.5013 |
| PA | Restaurants | 0.6526 | 0.5346 | 0.5877 |
| AROW | Restaurants | 0.4399 | 0.5423 | 0.4859 |
| L-BFGS | Laptops | 0.5969 | 0.3793 | 0.4639 |
| SGD | Laptops | 0.3357 | 0.3522 | 0.3438 |
| AP | Laptops | 0.5682 | 0.2463 | 0.3436 |
| PA | Laptops | 0.5935 | 0.4064 | 0.4825 |
| AROW | Laptops | 0.4349 | 0.3867 | 0.4094 |

Three other algorithms are implemented in CRF suite as well: Averaged perceptrons (AP), passive aggressive (PA), and Adaptive Regularization of Weight Vectors (AROW). Averaged perceptrons iterates over the training data, updating the feature weights of a perceptron whenever the model cannot make a correct prediction and updating the average feature weights. The final averaged feature weights are returned by the algori0074hm [20]. Passive-aggressive algorithms define a loss function on predicted instances, aggressively shifting the current parameter estimate when the current training instance has a positive value for the loss function and making no adjustment when the loss function is zero [21].AROW is a variation of confidence-weighted learning, which maintains a Gaussian distribution to measure the confidence in each parameter estimate. It adjusts the model to prevent overly aggressive shifts that can occur when using passive-aggressive updates [22].

## 5. Conclusion

Aspect based review system that identifies aspects dynamically is proposed in this paper. This proposed model considers Conditional Random Field which is a sequential labeling method to identify the aspects. The experiments are conducted on benchmark data sets and evaluating measures such as precision and recall shown significant improvement in the performance.

## References

[1] K. S. Jones. Natural language processing: a historical review. In Current Issues in Computational Linguistics: In Honour of Don Walker, pages 3{16. Springer, 1994

[2] N. R. Council, A. L. P. A. Committee, et al. Language and Machines: Computers in Translation and Linguistics; A Report. National Academy of Sciences, National Research Council, 1966

[3]J. J. Godfrey, E. C. Holliman, and J. McDaniel. SWITCHBOARD: Telephone speech corpus for research and development. In Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on Acoustics, Speech, and Signal Processing, volume 1, pages 517-520. IEEE, 1992

[4] L. Hirschman and R. Gaizauskas. Natural language question answering: the view from here. Natural Language Engineering, 7(04):275-300, 2001

[5] S. Bird. NLTK: the natural language toolkit. In Proceedings of the COLING/ACL on Interactive presentation sessions, pages 69{72. Association for Computational Linguistics, 2006.

[6] C. D. Manning, M. Surdeanu, J. Bauer, J. R. Finkel, S. Bethard, and D. McClosky.The Stanford CoreNLP Natural Language Processing Toolkit. In ACL System Demonstrations, pages 55-60, 2014.

[7] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19(2):313-330, 1993

[8] M. F. Porter. An algorithm for su_x stripping. Program, 14(3):130-137, 1980.

[9] M. P. Marcus, M. A. Marcinkiewicz, and B. Santorini. Building a large annotated corpus of English: The Penn Treebank. Computational Linguistics, 19(2):313-330, 1993

[10] G. A. Miller. WordNet: a lexical database for English. communications of the ACM, 38(11):39-41, 1995

[11] Sean Byrne, "Aspect Identification and Sentiment Analysis in Text-Based Reviews", Lehigh University, Thesis for Master of Science, 2017.

[12] T. N. Mansuy and R. J. Hilderman. Evaluating WordNet Features in Text Classication Models. In FLAIRS Conference, pages 568-573, 2006

[13] Pavlopoulos. Aspect based sentiment analysis. Athens University of Economics and Business, 2014.

[14] F. Sha and F. Pereira. Shallow parsing with conditional random fields. In Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1, pages 134-141. Association for Computational Linguistics, 2003

[15] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. Journal of the Royal Statistical Society. Series B (methodological), pages 1-38, 1977.

[16] C. Sutton and A. McCallum. An introduction to conditional random _elds. arXiv preprint arXiv:1011.4088, 2010

[17] Okazaki. CRFsuite: a fast implementation of conditional random fields (CRFs) 2007

[18] D. C. Liu and J. Nocedal. On the limited memory BFGS method for large scale optimization. Mathematical Programming, 45(1):503- 528, 1989

[19] S. Vishwanathan, N. N. Schraudolph, M. W. Schmidt, and K. P. Murphy. Accelerated training of conditional random fields with stochastic gradient methods. In Proceedings of the 23rd International Conference on Machine Learning, pages 969-976. ACM, 2006

[20] M. Collins. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10, pages 1-8. Association for Computational Linguistics, 2002

[21] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz, and Y. Singer. Online passive-aggressive algorithms. Journal of Machine Learning Research, 7(Mar):551-585,2006

[22] K. Crammer, A. Kulesza, and M. Dredze. Adaptive regularization of weight vectors. In Advances in Neural Information Processing Systems, pages 414-422, 2009