Enhance Sentence Level Sentiment Analysis on Review Text Data using Weighted Conjunction Analysis

Qing Wang†
Department of Statistics
University of Illinois Urbana Champaign
Urbana IL USA
qwang55@illinois.edu

Tianqi Wu†
Department of Industrial Engineering
University of Illinois Urbana Champaign
Urbana IL USA
twu38@illinois.edu

ABSTRACT

In this paper, we define and study a new rating analysis algorithm called Weighted Conjunction Analysis (WCA) for supervised sentiment analysis, which aims to tackle opinion shift in presence of conjunction words/phrases in multinomial sentiment classification tasks. We divide all conjunction words in four categories and build a WCA model that learns the weights of each side of each conjunction type using labeled reviews with only one conjunction. To make predictions on unlabeled reviews, we parse each review in presence of conjunctions (one or more) using a novel tree-based data structure called Logic Tree (LT) based on predefined conjunction priority and apply learned weights onto each logic tree to predict ratings. Empirical experiment on a Amazon fine food review dataset shows that WCA can successfully improve the performance of existing multinomial rating classification algorithms that do not involve conjunction handling. The results and weights mined by WCA for each conjunction type can also be used in a wide range of applications such as comparing reviewers' rating behaviors, improving ranking algorithms and refining recommender systems based on more accurate ratings.

CCS CONCEPTS

- **Information systems**→Data analytics; Sentiment analysis; Business intelligence;
- **Computing methodologies** → Natural language processing; Supervised learning by classification;

Keywords

Opinion and Sentiment Analysis, Conjunction Handling, Review mining

1. INTRODUCTION

With the rapid development of online shopping and reviewing websites, leaving online reviews and ratings for products is becoming a more and more common phenomenon for customers. As the volume of reviews becomes larger and larger, rating all reviews by browsing through every single review is extremely time consuming and expensive for human beings. As mining ratings from reviews becomes an unavoidable challenge, sentiment analysis has been proposed to study the opinion and emotion in text in 2001. However, there are many linguistic obstacles to accurately determine the emotion of a review such as negation handling and part of speech disambiguation.

A most common obstacle for sentiment analysis of compound sentence is conjunction handling as the presence of conjunction words can shift or negate the polarities of each part of a sentence with different magnitude. However, existing learning algorithms on multinomial rating analysis does not consider effect of conjunction in reviews. Existing study on conjunction handling by Arun and Prabhakar [1], Ritesh et al. [2], and Farooq [3] all focus on sentence level binomial sentiment classification with only positive or negative sentiment. Using their methods to handle the effect of conjunctions in rating analysis is not enough as most rating schemes have more than 2 levels such as level 1-5 or 1-10 rating schemes. Those conjunction handling methods are not capable of distinguishing reviews with same main sentiment (e.g. positive/negative) but different extent (e.g. strong or weak, rating of 1 vs 2).

Another deficiency of existing methods on conjunction handling is that they only focus on linguistic aspects but does not consider rating habits of reviewers for different field as there is no learning of labeled reviews involved. They define specific rules for each conjunction words only based on part of speech (POS) of words carrying polarities such as the polarity of the sentence is equal to the polarity of the right hand side of a conjunction word if words carry main polarity on both side are noun phrases. In addition, they only defined rules for sentences containing only one conjunction yet in reality, reviews containing more than one conjunction take up a large proportion. In order to apply conjunction handling to a wider range of data, it is also important to develop approaches to resolve conjunction handling in reviews with more than one conjunction.

In order to solve those issues, we proposed a novel algorithm call Weighted Conjunction Analysis (WCA) that not only considers linguistic aspects of conjunction words in a review, but also captures reviewers' rating behavior with model fitting process. We first divide all conjunction words and phrases into four categories based on criterions in Farooq's paper (2013), which are additive, contrasting, conditional and similarity conjunctions. Then we take all sentence level reviews with only one conjunction words as training data and parse the sentence based on conjunctions. To compute the polarity of each side, we use SentiWordNet library to calculate the average polarity of each word, sum them up and apply negation handling if needed using techniques discussed by Farooq (2017). Next, we fit a linear regression model with left hand side and right hand side polarities as predictors and ratings as response for each conjunction category and obtain fitted coefficients as that category's weights. For the prediction part that involves predicting reviews having multiple conjunctions, we proposed a novel tree structure called Logic Tree (LT) to divide each sentence by conjunction orders so that we can apply weights for single conjunction reviews on them to predict their ratings.

We evaluated our proposed algorithm with Amazon fine food and verified women e-commerce clothing reviews. The result shows that our algorithm achieves lowest root mean squared errors on ratings compared to existing supervised or unsupervised sentiment analysis algorithms as well as previous methods of conjunction analysis. The difference in weights learned from two datasets also shows that reviewers have difference rating behaviors in difference field and our method successfully captures that difference to yield accurate ratings. The result of WCA is meaningful as it quantifies influence caused by conjunction words in sentiment analysis on reviews so that the predicted ratings are more accurate. It also has a wide range of application including enhancing product ranking and recommender systems based on more accurate ratings. It can also be used to analyze reviewer behaviors in different fields so that different models can be used to predict ratings in each field such as clothing, dining, electric products and etc.

2. RELATED WORK

To our knowledge, no prior work has examined our proposed WCA and LG problem, but there are some related work.

Arun's paper [1] implements conjunction analysis which divides sentence into segments based on conjunction words and picks the best segment for sentiment analysis. Ritesh's paper [2] emphasis on sentence structure and conjunction analysis to find the phrase dominating the sentiment in a sentence. However, they only defined some basic conjunction rules and it is hard to work on more generalized situation. Farooq's paper [3] focuses on negation analysis which decides how to evaluate a sentence with negation words. Negation analysis is crucial since the presence of negation words would largely affect the sentiment of the whole sentence. Our study also includes a similar approach to consider the effect of negation.

Later, Farooq et al [4] proposed conjunction classification and rules dealing with conjunctions of these classes to determine the polarity at both the sentence level but also at the feature level. The conditional conjunctions introduced is useful indeed but it has some deficiency. Several conjunction rules were defined depending on part of speech (POS) but the model fitted could not distinguish it well. For example, the method proposed could not tell difference between noun "love" and verb "love". This kind of ambiguation, however, is quite common in reality and it will certain affect the classification result. With the attempt of WCA, our rules does not depend on POS but only conjunction types, which is much easier to implement in reality.

In addition to the deficiency discussed above, there are several common flaws that related work shares. Most of the existing methods assume that with the presence of conjunction, not all parts of a sentence contribute to the sentence's polarity. It is enough to only consider the polarity of specific segment of a sentence based on predefined rules. However, it is also important to analyze the parts that do not contribute to main polarity. Those parts might increase or diminish the strength of main polarity sometimes. Moreover, none of the previous work consider the situation that multiple conjunction words appear in one sentence, which happens most of the time.

Previous study may work well if we only need to do binary classification (positive or negative sentiment). If more specific rating scheme (1-5 or 1-10) is given, new method is required to achieve better classification results. In this study, we focus on this problem and WCA is a novel way of conjunction words analysis where instead of discarding the unimportant segments of a review, we fit sentiment analysis algorithm on each segment and then assign weights to each segment's rating to produce overall rating. Problem of multiple conjunction words is also handled by introducing the novel Logic Tree structure (LT).

3. PROBLEM DEFINITION

In this section, we formally define terminologies we use for Weighted Conjunction Analysis (WCA).

First, we assume the input of our algorithm is a set of sentence level reviews containing conjunctions with labeled overall ratings where its size is constantly expanding (getting more reviews). We denote this set as $D = \{d_1, d_2, ..., d_{|D|}\}$. We also use D_1 to denote the subset that contains all reviews with only one conjunction word. We use $V = \{w_1, w_2, \ldots, w_n\}$ to denote the vocabulary of all unique words in D.

Definition (Overall Rating) An overall r_d of a review d is an ordinal numerical measure of reviewer's overall opinion towards a subject. For both datasets in our study, r_d is a rating in $\{1, 2, 3, 4, 5\}$ with $r_d = 5$ indicating very satisfied and $r_d = 1$ indicating very unsatisfied.

Definition (Conjunction Term) A conjunction term is any uninflected linguistic form that connects words, phrases and clauses in a sentence. We use $C = \{c_1, c_2, \ldots, c_{|C|}\}$ to denote the set of all conjunction terms presented in D.

Definition (Conjunction Category) We divide C into four categories based on their functionalities as conjunction terms — additive, contrasting, conditional and comparative similarity. We use $C_{additive}$, $C_{contrasting}$, $C_{conditional}$ and $C_{similarity}$ to denote all additive, contrasting, conditional and comparative similarity conjunction terms presented in D. Table 1 shows examples of each subset.

Table 1 Types of Conjunctions and Examples

Type	Example
Additive	And, moreover, also, as well as, also
Contrasting	But, however, instead, except, although
Conditional	If, unless, as long as, when
Comparative	Or, similarly, equally, likewise

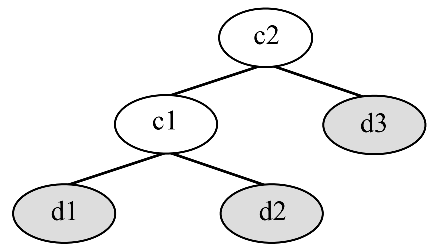
Definition (Polarity) In our study, the polarity $P(w)$ of a word w is a measure of the emotion expressed in the word. It can range from -1 to 1 with -1 being extremely negative emotion, 1 being extremely positive emotion and 0 being neutral emotion. For sentence level polarity, we use $P(S)$ to denote the polarity of sentence S. Table 2 shows examples of words and their average polarities from SentiWordNet library.

Table 2 Sample Word Polarities

Word	Polarity
Nice	0.8125
Ridiculous	-0.4375
Recommend	0.4375
Asian	0.0

Definition (Logic Tree) A logic tree T is a tree decomposition of a sentence with all nodes being conjunction terms and all leaves being segment of sentence. For sentence $S = (d_1, c_1, d_2, c_2, d_3)$ containing two conjunctions c_1, c_2 with priority $c_2 > c_1$ and three segments d_1, d_2, d_3, a logic tree T_S of S will be:

Figure 1 Logic Tree Demo



Definition (Weighted Conjunction Analysis) Given a dataset D where each review d is associated with a rating r_d and a set of all unique conjunction terms C divided into $C_{additive}$, $C_{contrasting}$, $C_{conditional}$ and $C_{similarity}$, taking a subset D_1 of D with reviews containing only one conjunction term, for a review d_1 in D_1 containing conjunction term c, we parse d_1 by c and denote it as (d_{LS}, c, d_{RS}) and calculate segment polarities $P(d_{LS})$ and $P(d_{RS})$ by predefined polarities. Weighted Conjunction Analysis then finds the weights β_{LS}, β_{RS} for $P(d_{LS})$, $P(d_{RS})$ with labeled rating r_d for c in each conjunction category and predict ratings for remaining reviews in D based on the weights for each conjunction category and logic tree T for each d.

4. METHODS

WCA is neither completely supervised nor completely unsupervised algorithm. The determination of each segment's polarity is unsupervised using SentiWordNet library and negation handling, while the process of obtaining conjunction weights is supervised. In this section, we introduce techniques we adopt to calculate polarities as well as statistical models and data structures we use to fit and predict final ratings.

4.1 Segment Polarity Determination

The objective of this step is to develop a systematic approach to calculate polarities of sentence segments so that it can be used to generate training data for WCA.

4.1.1 Word Polarity Determination

The first challenge is which polarity determination method to use in order to best achieve our research purpose. There are many existing methods that calculate sentiment of multiple words including supervised and unsupervised approaches. We finally decide to stick to unsupervised approach using SentiWordNet library where each word has a polarity score for each of its senses. And we only consider the polarities of nouns, verbs, adjectives and adverbs in a segment. We make this decision because from our experiments we find that using supervised polarity determination only improve model performance by a small amount but is much more time-consuming. As our algorithm aims to provide improvement for large review datasets, unsupervised polarity determination is much easier to implement in reality and its performance is almost as good as supervised polarity determination if we use a powerful word sentiment library e.g. SentiWordNet.

4.1.2 Negation and Word Sense Handling

Another challenge is to consider linguistic effects of negation and word sense ambiguation in unsupervised polarity determination. Negation words can significantly affect a segment's polarity and the presence of word sense ambiguation can make polarity determination process more unreliable. To solve the negation problem, we adopt the most recent negation handling approach proposed by Farooq's paper [3] where when encountering a negation term, we negate the polarity of words after it until the next conjunction term, comma or period, whichever comes first. To solve word sense ambiguation for a word, we use the average of polarity scores of all senses as its final polarity. We decide to use average polarity because from our experiments, existing word sense disambiguation approaches still yield many misclassified senses and result in similar performance as the average approach. Another reason is that using average approach takes much less time and effort in reality.

Finally we find the polarity of a segment using the sum of scores of all words in that segment. Figure 2 demonstrates our polarity determination approach:

Figure 2 Polarity Determination Demo

"I love the material of this dress but do not like the color."
S1 = "I love the material of this dress" *P(S1) = P(love) + P(material) + P(dress) = 0.8847*
S2 = "don't like the color" *P(S2) = - (P(like) + P(color)) = - 0.6597*

4.2 Weighted Conjunction Analysis

In the second stage, we used polarity determination method defined in 4.1 to generate training data for weighted conjunction model and use linear regression to find weights for each conjunction type.

4.2.1 Generation Assumption

We make following assumptions on a reviewer's behavior of using conjunctions.

- *Contrasting Conjunction*: to express an opposite emotion of previous segment.
- *Additive Conjunction*: to express a complementing emotion of previous segment.
- *Conditional Conjunction*: to diminish the emotion of previous segment as the emotion is limited to certain conditions.
- *Comparative Conjunction*: to express same emotion of previous segment.

4.2.2 Linear Regression Model

In training phase, we use only reviews containing only one conjunction term as our training data but no reviews containing more than one conjunctions. The reason is that those reviews directly relate each segment's polarity with overall ratings so that it conveys most information about how conjunction words affect the polarity of segment on each side. We denote such review dataset as D_1. Next, we parse each review in D_1 by conjunction term into LS (left-hand-side of conjunction) and RS (right-hand-side of conjunction) and apply polarity determination approach on LS and RS to obtain $P(LS)$ and $P(RS)$ for each review. We then divide those polarities into 4 subsets based on types of conjunction terms and generate training data X for each conjunction type. The response Y is just the ratings for all reviews in D_1 scaled to [-1, 1] for better model fitting. Now, for each conjunction type, we can formulate our linear regression model with following equation:

$$Y = X\beta + \epsilon \tag{1}$$

where X is a $(n \times 3)$ matrix with first column being all 1's, second column containing all $P(LS)$'s and third column containing all $P(RS)$'s for that conjunction type. Y is a $(n \times 1)$ vector containing ratings for corresponding reviews. β is a (3×1) vector with first element being fitted intercept, second element being fitted weight for $P(LS)$ and third element being fitted weight for $P(RS)$.

We then use Least Square Estimation with following formula to obtain fitted β for each conjunction type:

$$\hat{\beta} = (X^T X)^{-1} X^T Y \tag{2}$$

Algorithm 1 on next page illustrates detailed steps for WCA. With obtained fitted weights, we can move onto rating prediction part.

4.3 Rating Prediction with Logic Tree

In this stage, we aims to predict ratings for reviews including one or more conjunction words using fitted weights from WCA for each conjunction category.

4.3.1 Logic Tree Structure

For reviews with only one conjunction word, we just apply fitted weights onto each segment's polarity to obtain final ratings. A challenge to predict ratings for reviews containing multiple conjunction words is that directly applying weights on each segment is not applicable as one conjunction word may affect multiple

Algorithm 1: Weighted Conjunction Analysis

Input: a collection of reviews containing only one conjunction D_1, a collection of their scaled ratings Y, and 4 sets of conjunction terms categorized by type $C_{additive}$, $C_{contrasting}$, $C_{conditional}$ and $C_{similarity}$.
Output: 4 fitted weight vectors for $P(LS)$ and $P(RS)$, $\hat{\beta}_{additive}$, $\hat{\beta}_{contrasting}$, $\hat{\beta}_{additive}$ and $\hat{\beta}_{additive}$.

***Step 0**:* Divide all reviews into 4 subsets based on conjunction type sets;
For each subset, do step 1-4:
***Step 1**:* Parse each review d into LS and RS;
***Step 2**:* Find $P(LS)$ and $P(RS)$ for each review d and store them as rows in matrix X;
***Step 3**:* Find least square estimate $\hat{\beta}$ of β in linear regression model $Y = X\beta + \epsilon$;
***Step 4**:* Store $\hat{\beta}$ as fitted weights for corresponding conjunction type;
***Step 5**:* Output obtained $\hat{\beta}_{additive}$, $\hat{\beta}_{contrasting}$, $\hat{\beta}_{additive}$ and $\hat{\beta}_{additive}$;

segments in the sentence. To resolve this problem, we proposed a novel data structure Logic Tree (LT) where each segment is stored as a leaf node and each conjunction term as a node in a tree structure based on conjunction priority. We then apply fitted weights from leaf nodes to root node of that tree to predict final rating. After investigating a portion of reviews, we find a reasonable priority order of conjunction types being: contrasting > conditional > additive > similarity. The more prior a conjunction type is, the more hierarchy it has in a logic tree. The detailed steps of building logic tree of a review is shown in algorithm 2.

Algorithm 2: Building Logic Tree

Input: a review with conjunction word d, a list l of priority order of conjunction types.
Output: a logic tree object *T* of review *d.*

***Step 0**:* Find the most prior conjunction term in *d* based on *l* and make it as current root value;
***Step 1**:* Parse d into LS and RS and store them as left child and right child.
***Step 2**:* For each child,
if (it does not contain any conjunction term):
    return the entire segment.
else:
    repeat step 0-1 using recursion on that node until there is no more conjunction term in any leaf node.
***Step 3**:* Output obtained logic tree *T*.

Then, we can apply fitted weights on logic tree to perform rating prediction for reviews with multiple conjunctions.

4.3.2 Rating Prediction

With weights and logic trees obtained from previous stages, the rating prediction process is quite straight forward. For review with one conjunction, we apply fitted weights directly to segment on each side. To make predictions on a logic tree with multiple conjunction terms, we first replace each segment in leaf node with its polarity. Then, we do a traverse from leaf to node while replacing conjunction term in parent node by polarity calculated from applying fitted weights onto each child of it. Algorithm 3 illustrates the process in detail.

Algorithm 3: Rating Prediction with Logic Tree

Input: a logic tree object *T*, 4 pairs of fitted weights of each conjunction type.
Output: a scaled rating *r* for *T.*

From root node do following recursion:
Recursion*:*
if (current node does not have child):
    return segment polarity *P(node)*;
else:
    denote fitted $\hat{\beta}$ of current node's conjunction term's conjunction type as $(\beta_{intercept}, \beta_{LS}, \beta_{LS})$, we calculate current node's polarity with
    $P(node) = \beta_{intercept} + \beta_{LS} P(child.left) + \beta_{RS} P(child.right)$
    where *P(child.left)* and *P(child.right)* are calculated by recursion.

Finally, we scale back all predicted ratings to yield final result.

5. EXPERIMENT RESULTS

In this section, we first introduce the review data sets used for experiments applying WCA. Also, discussion of the experimental results are included.

5.1 Experimental and settings

To test the effectiveness of Weighted Conjunction Algorithm that we proposed, experiments on two data sets are performed. The first one is *Women E-Commerce Clothing Reviews* from Kaggle. We chose this data set because it is relatively small and suitable for preliminary experiment. We use this data set mainly for weights comparison purpose. Reviewers provide ratings on the cloths from 1 Worst, to 5 Best. There are 23486 reviews in total and we find that 1297 of them have conjunction words. We only keep variable *review text* and *rating* for testing and examples are given in Table 3. The data is available at https://www.kaggle.com/nicapotato/womens-ecommerce-clothing-reviews.

The second data set that we used is *Amazon fine food review.* This data set is much larger than the first one and can be used to validate the performance of our method. Most of the experiment and evaluations are focused on this data set. There are 393579 reviews and 36080 of them are sentence level reviews having conjunction words. We use those reviews as our dataset with The rating scheme also has 5 levels and we only keep variable *text* and *score.* The data is available at https://www.kaggle.com/abhinandan103/amazon-fine-food-review. Preprocessing of the data sets includes converting the words to lower cases and removing the punctuations in sentences.

Table 3 Clothing Review Examples

Review Text	Rating
Looks beautiful online but has too much material and the zipper catches on the lace.	3
Love this sweater, do wish I sized up for a little longer length, but works great with high rise pants.	4
I really wanted this skirt to work but it didn't look very flattering on me.	1
…	…

5.2 Qualitative evaluation

In this section, we will show two sample results for our proposed model for qualitative evaluation.

First, we provide examples for sample Amazon fine food reviews with observed and fitted ratings. The results are shown in Table 4. In general, our model predicts better for reviews with higher observed ratings (4-5). It is expected since there are much more reviews with higher observed ratings in the data set. It means that there is much more training data for the model to learn. Overall, the fitted ratings resulted from our model are close to the observed ones.

Table 4 Sample Review

Review Text	Obs. Rating	Fitted Rating
i like these better than the regular altoids but theyre even more costly so youd better like them	4	4.3
great product nice combination of chocolates and perfect size the bags had plenty and they were shipped promptly the kids in the neighborhood liked our candies	5	4.8
i dont plan on reordering it but it was definitely edible and filling since i added chicken	3	3.2
i dont know whether it was a fault in the packaging or if they were just past their prime but they were rock hard and the flavor was not strong enough	2	2.9

Then, we want to further analyze the difference of weights among the four conjunction types. The fitted coefficients of segments on each side of each conjunction in two data sets are shown in Table 5, where comma separates the weights on the left and right of the conjunctions. For Amazon fine food data set, it can be seen that the weight of left segment is larger than the right one for *contrasting* and *conditional conjunctions.* The larger magnitude of weight for *contrasting conjunction* indicates stronger effect on sentence clarity than *conditional conjunction*. The weight of left and right segment is close for *additive* and *similarity conjunctions.* The larger magnitude of weight for *additive conjunction* indicates that stronger effect on sentence polarity than *comparative conjunction*. Also, the weights for two data sets are quite different, which indicates that reviewers' rating behavior varies among domain of study. Conjunction effects on rating are different in different fields and we should not apply the same weights of conjunction types for all the data sets.

Table 5 Weights for Conjunctions

Conjunctions	Women Clothing	Amz. Fine Food
Additive	0.0056, 0.0092	0.0105, 0.0092
Contrasting	0.0038, 0.0313	0.0219, 0.0040
Condition	-0.0150, 0.0872	0.0130, 0.0045
Similarity	-0.0218, 0.0127	0.0100, 0.0074

5.3 Quantitative evaluation

We use the logistic regression from python package *scikit-learn* as our baseline model. For quantitative evaluation, we will use RMSE as our performance measure. It is calculated as below:

$$RMSE = \sqrt{\frac{1}{n}\Sigma_{i=1}^{n}\left(p_i - o_i\right)^2} \tag{3}$$

where p_i is the predicted rating , o_i is the observed rating and n is number of observations. The results are shown in Table 6, where the percentage in the bracket indicates the fraction of data that we use for training. Women clothing data set has lower RMSE overall because it has much smaller average review length. Our algorithm calculates segment polarity by summation. Therefore, the total prediction error in polarity estimation for all words is smaller for shorter reviews.

Table 6 RMSE of Mehods

Method	Women Clothing	Amz. Fine Food
supervised(10%) without conjunction handling	1.1450	1.1858
supervised(50%) without conjunction handling	1.0708	0.9971
Unsupervised without conjunction handling	1.1448	1.2278
WCA (10%)	1.0249	1.1253

We make these comparisons since only small proportion of sentence level reviews have only one conjunction term so our algorithm only use 10% of the dataset as training data. It is expected that the performance would improve if the model has more labeled data to learn. For both data sets, it is clear that WCA performs better than unsupervised and supervised (10%) logistic regression model without conjunction handling. Analysis of conjunction is indeed important in this case. Our proposed method also has shorter running time overall.

Supervised (50%) logistic regression model also has very low RMSE on for both data sets. The reason is that it uses 50% of data in training stage. This is an advantage as well as disadvantage of completely supervised algorithms, where it needs relatively large portion of training data to achieve better performance than other algorithms. However, with rapidly expanding review data sets in reality, it is extremely time consuming to use 50% as training data and fit models constantly. Our model is able to achieve very close performance to supervised (50%) logistic regression model with only 10% training data. The reason is that we do not rely on supervised algorithms to find vocabulary polarities but powerful sentiment analysis libraries such as SentiWordNet. We only need supervised model to capture influence of conjunction terms on sentence polarity, which is definitely a more efficient choice in reality.

5. CONCLUSIONS

In this paper, we discussed a novel rating analysis algorithm called Weighted Conjunction Analysis (WCA). It uses linear regression to find polarity weights separated by four specific conjunction types. To solve the problem of presence of multiple conjunctions in reviews, we introduced a novel tree structure called Logic Tree (LT), which uses the fitted weights from WCA to make predictions. Combining the two methods, we are able to tackle multinomial sentiment classification tasks with the presence of conjunctions. Given review text and corresponding ratings, it makes predictions based on supervised learning of conjunction weights. Experimental results on women clothing and Amazon fine food review data sets show that our proposed methods can effectively improve the performance of prediction given the same proportion of training data.

For application aspect, our work can be used as a complement of any sentence level supervised sentiment analysis algorithm in the presence of conjunction with rating scheme of more than 2 levels. It also provides a novel direction of dealing with multiple conjunctions. However, since the conjunction weights would vary among different domains, it is impossible to set the same weights for all data sets. Learning of labeled data is required in this case.

7. ACKNOWLEDGMENTS

We want to thank the reviewers from two data sets for their valuable comments and Kaggle for presenting the verified data sets. This study cannot be done without the instruction and suggestion given by Professor C, Zhai and TA X, Liu. Both members in the team participate in this project actively and substantially. Tianqi Wu investigated related work and Qing Wang defined the problem. We found the solutions and did experiments together.

8. REFERENCES

[1] Arun, M., & Prabhakar, T. V. (2007). Sentence Level Sentiment Analysis in the Presence of Conjuncts Using Linguistic Analysis. *Lecture Notes in Computer Science Advances in Information Retrieval,* 573-580.

[2] Ritesh, A., Prabhakar, T. V., & Chakrabarty, S. (2008). "I Know What You Feel": Analyzing the Role of Conjunctions in Automatic Sentiment Analysis. *Advances in Natural Language Processing Lecture Notes in Computer Science,* 28-39.

[3] Farooq, U, Antoine, N., & Muhammad, A. (2016). Negation Handling in Sentiment Analysis at Sentence Level. *Journal of Computers, 12*(5), 470-478..

[4] Farooq, U., Nongaillard, A., Ouzrout, Y., & Qadir, M. A. (2013). Product reputation evaluation: the impact of conjunction on sentiment analysis. Proceedings of 7th Intl Conference on Software, knowledge, Information Management and Applications (SKIMA). (pp. 590–602).