Abstract—Sentiment classification has been an essential part of opinion mining and sentiment analysis. This topic has been applied to real world scenarios such as mining customer reviews on merchandise sold online and film reviews of movies. Therefore, we aimed to gain insight into sentiment word classification, as it could serve as the foundation for larger scale sentiment analyses on corporuses and documents. In this paper, we focus on word polarity classification, which could be extended to perform classification of sentences and paragraphs. We enhanced our previous work on gloss vector and expanded it with a more concise method to generate the vectors. Additionally, we used more sources to validate the similarities of the candidates with two vectors, each representing the positive and negative sentiment polarity respectively by importing groups of words that express that polarity. Experiment results demonstrated that our method is effective, while producing better accuracies than the previous attempt on similar subjects.

Keywords—sentiment analysis; word polarity classification; sentiment sensitive vector; lexical taxonomy; word similarity

I. INTRODUCTION

With the explosion of Web 2.0 services, more and more user-generated sentiment data have been shared on the Web. They exist in the form of user reviews on shopping or opinion sites, posts of blogs or customer feedbacks. It is useful for both consumers as well as business corporations to know what the public generally thinks about a particular product or service. As a result, opinion mining has become a topic that received much attention recently [1][2], with examples including opinion summarization [3][4], opinion integration [5] and review spam identification [6]. Sentiment classification, which aims at classifying sentiment data into polarity categories (e.g., positive or negative), is widely studied because many users do not explicitly indicate their sentiment polarity, and it can only be identified from user-generated text data. For instance, a sentiment classifier may classify a film review about a movie as positive or negative depending on the sentiment expressed in the review.

The crux of performing such classification is the recognition of sentiment carrying words in a sentence. Adjectives were initially considered as the primary sentiment carriers in a document [3][13][14]. Eventually, more exceptions have been discovered, wherein other parts of speech are considered helpful in sentiment mining. These also include the use of adverbs [15]; adjectives and verbs[10]; adjective phrases[16]; two-word phrases [17][18]; adjectives, verbs, and adverbs [19]; the exclusive use of verbs [20]; the use of non-affective adjectives and adverbs [21]; or rationales, words and phrases selected by human annotators [22]. Once the sentiment carriers are identified, the next step is to determine the polarity of the sentiments, i.e., whether the carrier represents a positive or negative sentiment. There are two approaches to solve this problem. The first involves creating a dictionary with words annotated with their semantic orientation (polarity and strength) and incorporating the relative intensification and negation [23]. This dictionary is then looked up to find the sentiment orientation of words. The major disadvantage of this technique is the high cost of constructing a sentiment dictionary. Hence, a second approach that reduces labor cost through automated processes is proposed. In this method, sentiment of words is identified through machine learning techniques [24].

One way of clustering similar sentiment carriers is through the use of WordNet\(^1\), a large lexical database from the Cognitive Science Laboratory of Princeton University that also provides relationship information among concepts. Semantic similarity between nouns and verbs can be easily evaluated from WordNet using the taxonomic relations. Taking Figure 1 as an example, the semantic similarity between any two node scan be computed by counting the distance to the nearest common node. We can easily infer that the similarity between snake and crocodile will be greater than that between snake and elephant, judging by the smaller distance that we have to trace to the common node.

![Figure 1: Taxonomy of example nouns in WordNet.](https://wordnet.princeton.edu/)

---

\(^1\)https://wordnet.princeton.edu/
However, hierarchical structure of such for adjectives and adverbs does not exist in Wordnet. In addition, by looking into data from the Internet, we discovered that it is more likely to find average users using simple words to express their emotions than using intricate and carefully crafted phrases. To illustrate, we gathered some example texts from the user reviews of Amazon and listed them in Figure 2. As we can see, most of the comments we find on the web contain at least one part-of-speech that delivers a certain level of sentiment. Considering that adjectives and adverbs are frequent sentiment carriers, we emphasize on the extraction of information regarding these two POSs from WordNet. The keystone to our approach is that the sentiment orientation of a word w can be found by comparing its semantic similarity with words representative of both the positive and negative sentiment class, respectively.

Since sentiment dictionaries are widely used to capture the polarity of words for sentiment analysis, novel and unseen events are the most challenging issues of this task. To address this problem, we can classify the polarity of unseen words through the calculation of word similarity. In this paper, we proposed a sentiment sensitive vector for word polarity classification, which is an automated process that integrates various gloss of word to generate the sentiment sensitive vector, or SSV, for word representation. Our experiments demonstrated that the proposed method is effective in word polarity classification.

The remainder of this paper is organized as follows. Related works are reviewed in the next section. We elucidate our concept and methodology in Section 3, which includes the generation of gloss vectors from WordNet and the metrics we use for evaluating similarity. The actual procedures of experiments and corresponding results are shown in Section 4. Finally, we bring in our conclusion in Section 5.

II. RELATED WORK

Articles are one of the most common ways for persons to convey their feelings. Identifying essential factors that affect sentiment transition is important for human language understanding. With the rapid growth of computer mediated communication applications, such as social websites and microblogs, sentiment analysis has been ported to a wide spectrum of applications. Several researchers performed this practice, one of them being the work of Hu and Liu [3] on mining customer reviews via the usage of sentiment dictionaries. They started with tagging POS for words in paragraphs of text, followed by determining the frequent features being discussed for a merchandise. Opinion words are then extracted to predict the orientations of opinion sentences regarding these features, which later lead to the summarization process. Their work was based on previous studies [27][28], which constructed a dataset for the use of subjectivity classification and its implementation on sentence-level manual tagging, and had already established a positive statistically significant correlation between polarity and the presence of adjectives. Therefore, utilizing dictionaries with information regarding polarities turns out to be an intuitive way of conducting sentiment classification. Nevertheless, when a corpus encounters an unseen candidate, measures have to be taken to accommodate the “new comer”. One of the methods frequently used in information retrieval is performing word expansion. Still, in situations where node-based or edge-based approaches are not applicable, other types of solutions need to be adopted. Consequently, we put our emphasis on the accurate and versatile classification of word polarity.

Word polarity classification has been widely investigated in the past few years. Most of these studies are based on publicly available language resources to recognize the polarity of words. For example, Turney and Littman [29] proposed an unsupervised algorithm in which they defined seven positive and seven negative paradigm seed words. They used the English web corpus to query any given word with the paradigm words by using the near operator in a search engine. If the word tends to co-occur with positive paradigm words, it is classified as positive. Otherwise, it is considered as negative. Patwardhan [8] extended context vector with the concept of gloss vector to perform disambiguation, which is practically a context vector generated by treating the glossary of WordNet as the context. By looking into the gloss of a certain word and computing the word vectors associated with those keywords, one could obtain the gloss vector and use it to compare the similarities between different sentiment polarities. Moreover, Takamura et al. [30] proposed a method that regards semantic orientation as spin of electrons. They consider each word as an electron and its polarity as a spin value. A word relatedness graph is constructed by using the gloss definitions, thesaurus, and co-occurrence statistics for English, and words are classified as either positive or negative according to their spin values. Kim and Hovy [10] evaluated the sentiment of an opinion holder by exploiting WordNet to generate lists of positive and negative words by expanding seed lists. They assume that the synonyms of a word would have the same polarity. The percentage of a word’s synonyms belonging to lists of either polarity was used as a measure of its polarity strength, while those below a threshold were deemed neutral or ambiguous. Their best results were achieved when the topic neighborhood consisted of words between the topics up to the end of the sentence. Hassan and Radev [31] introduced a semi-supervised method where a random walk model was used to find the polarities of English words. They assembled a word relatedness graph by using the relations in English WordNet, and manipulated mean hitting time for polarity estimation.

In light of the importance of word polarity classification, we propose an effective approach for this topic. Our approach differs from existing word polarity classification methods in a
number of aspects. First, we generate the gloss vector for a candidate word to be tested with WordNet glossary as the main corpus. Second, we use two lists of keywords that represent positive and negative polarities to generate two sentiment sensitive vectors for comparison with the candidate words. During this process, the 5-fold cross validation is used to provide a more evenly distributed data set and result. Finally, we calculate the semantic similarities of the candidate word with the two gloss vectors by using cosine similarity, and decide its polarity with the higher value of the two.

III. METHODOLOGY

The goal of sentiment word classification is to determine the most appropriate sentiment polarity (i.e. positive or negative) for a word $w$. In this paper, we use WordNet to construct sentiment sensitive vectors (SSV) for the representation of positive and negative polarities. Word sentiment polarities are then recognized by measuring the similarities between vectors. Our system consists of three main components, namely WordNet, GlossVectorCreator, and SemanticClassLabeler as shown in Figure 3. Training word sets are fed into the WordNet database to extract the glosses, from which the SSVs are generated by using the Gloss Vector Creator. The gloss vectors, as well as the training word lists, will later serve as the input to Semantic Class Labeler, in which the actual polarity classification is performed. Details of these components will be elaborated in the following sections.

A. Constructing Sentiment Sensitive Vectors from WordNet

To illustrate the process of constructing sentiment sensitive vectors from WordNet, we begin with the attempt of implementing gloss vectors to the original glossary of WordNet from which they were generated. Take Figure 4 as an example, in which we want to generate the gloss vector for “excellent” with the definition of “very good; of the highest quality”. First, we remove the stop-words from the gloss and lemmatize the remaining terms to acquire the list: “good, high, quality”. Subsequently, we search through the whole glossary of WordNet to look for the occurrence for each of these terms, whereas the occurrence is the value of dimension in the vector.

![Fig. 3. System architecture.](image)

![Fig. 4. Using WordNet to generate gloss vector of excellent.](image)

When the term $good$ shows up in the gloss of another word like terrific, we include all other non-stopwords that are present in that gloss (shown in blue font). Finally, since terrific is synonymous with “extraordinarily good or great” in WordNet, we add the terms extraordinarily and great into the vector. If the term is already in the gloss vector, we increment the value and accumulate it with each match. The same procedure is then applied to the rest of terms.

In summary, the gloss vector is generated through a two-level sweep of WordNet. By looking up the gloss of the candidate word, we completed the first-order indexing. Next, we use this result to find the co-occurrence of other terms through the glossary, which is the second-order sweep. Unfortunately, after initial experiments we realized that by conducting a second-order sweep, not only the gloss vector would grow to an enormous size, but it would also include a great amount of words that are not suitable as identifiers for sentiment classification. To resolve this problem, we have decided to neglect the second-
order sweep of the corpus, and use only the first-order keywords obtained from the gloss of each word. Regarding the example in Figure 4, we would therefore use the words good, high, and quality. The SSV is then built with gloss vector as the base, in which the positive and negative sentiment sensitive vectors are denoted as $SSV_p$ and $SSV_n$, respectively. The process of SSV construction is illustrated in Figure 5.

To construct a SSV, a list of keywords with predetermined polarities is used. A gloss vector is generated for every word in the list, and these gloss vectors are then combined together by including all dimensions into one vector, forming the sentiment sensitive vector. Lastly, a 5-fold cross validation is performed on the word list, in which we take out one fold as the testing set and the remaining four as the training set. The words in the training set are used to generate $SSV_p$ and $SSV_n$, which are later used to validate every candidate word in the testing set. Each candidate word will then acquire one similarity value with each SSV.

B. Measuring Similarity for Word Polarity Classification

During the measurement of similarity for word polarity classification, each candidate word $w_i$ will acquire two similarity scores by calculating cosine similarity with both the positive and negative SSVs, respectively. The polarity of word is defined in (1):

$$\text{Polarity}_{w_i} = \arg \max_{SSV \in SSV_p, SSV_n} \text{Sim}(w_i, SSV) = \frac{\sum \bar{V}(w_i) \cdot \bar{V}(SSV)}{\left| \sum \bar{V}(w_i) \right| \left| \sum \bar{V}(SSV) \right|}$$

(1)

where gt is the gloss term in gloss vector $V(w_i)$ and sgt is the sentiment gloss term of the sentiment sensitive vector SSV. The numerator is the dot product of the two vectors, whereas the denominator is the multiplication of the dimension. If two vectors are completely aligned, the similarity would be 1.0. The higher of the two resulting values decides the sentiment polarity of the candidate.

IV. EXPERIMENT

A. Experiment Setup

In order to employ a collection of vocabularies that can serve as a good indication of polarities, we took the sentiment dictionary from Liu et al. [11] featured in their sentiment analysis experiment on social media as the input corpus. The corpus is composed of positive and negative parts containing approximately 2000 and 4500 words, respectively. To limit the factors that would potentially affect the outcome, we categorized the lists into four part-of-speeches. We noticed that during the training process, only words of the same POS as the testing set would be considered when constructing the sentiment sensitive vectors. Also, in the original lists, misspelled words were intentionally added to represent the
typos that frequently appeared in social media content. These words and other POSs were left out of the sets. The distribution of words is illustrated in Figure 6. The system is developed under the .NET platform with external C# API from Matt Gerber of Michigan State University to extract the glosses from WordNet with the latest version of database file (Version 3.1). Additional codes were later constructed to create the gloss vectors, separate the word lists into folds for cross validation, and so on.

To evaluate the effectiveness of the system, we adopted certain accuracy measures. After computing the cosine similarity between the candidate word and both sentiment sensitive vectors obtained from the training set, we take the higher of the two values as the resultant polarity. Since we are using words with clear distinction on the sentiment, thresholds for the similarity was not used. That is, we did not have an additional “neutral” class beyond the “positive” and “negative” classes. Then, for each hold-out of the 5-fold method, the accuracy is computed as the number of words correctly classified under the source polarity out of the total number of words contained in that fold of polarity list. For example, in the first fold of adjective test, we have 143 out of 168 words classified under the positive class. Accuracies for this fold were therefore 85.12% and 71.25% for the positive and negative classes, respectively. Lastly, we used the macro-average and micro-average to compute the average performance for the five hold-outs. These measures are defined based on a contingency table of predictions for a target emotion $E_k$. The accuracy $A(E_k)$, macro-average $A^M$, and micro-average $A^\mu$ are defined as follows:

$$A(E_k) = \frac{TP(E_k) + TN(E_k)}{TP(E_k) + FP(E_k) + TN(E_k) + FN(E_k)}$$

(2)

$$A^M = \frac{1}{m} \sum_{k=1}^{m} A(E_k)$$

(3)
\[ A^\mu = \frac{\sum_{k=1}^{m} TP(E_k) + TN(E_k)}{\sum_{k=1}^{m} (TP(E_k) + FP(E_k) + TN(E_k) + FN(E_k))} \]  

where \( TP(E_k) \) is the set of test documents correctly classified to the emotion \( E_k \), \( FP(E_k) \) is the set of test documents incorrectly classified to the emotion, \( FN(E_k) \) is the set of test documents wrongly rejected, and \( TN(E_k) \) is the set of test documents correctly rejected.

B. Results and Discussion

Experiments were conducted on all four POSs with results shown in TABLE I. Verbs achieved the highest overall accuracy of 77.62%, with 64.58% for positive class and 81.68% for negative class, respectively. We can also see that apart from adjectives, most POSs have higher accuracies in the negative category, potentially owing to the higher number of words compared to the positive category, which results in a wider and more diverse data set. It is worth noting that in the category of adverb, accuracies among the two groups showed a substantial difference between 28.73% and 80.7%. One of the reasons for this would be the smaller amount of words in the training set for adverbs. While a total of 2480, 2165 and 1303 words are available in the lists of adjective, noun and verb, respectively, only 726 words exist in the adverb list. This may also due to the reason that the components forming the negative gloss vector in adverbs contain more general terms that are abundant in the glosses of words in both polarities. On the other hand, the positive group achieved an accuracy of 82.41% while the negative group acquired 67.68% for adjectives. After weighting both with their amount of data, a total of 72.68% accuracy was obtained. In addition, nouns achieved an average of 72.83% in accuracy. Adjectives and nouns account for nearly 70% of the total input, thus it’s not surprising that the overall micro average could achieve 72.42%.

![Fig. 7: Dispersion of similarity points of part-of-speeches.](image-url)
We also visualized the dispersion of data points belonging to different POSs among the space of similarity between positive and negative gloss vectors. As shown in Figure 7, the X-axis represents the similarity between the word and positive SSV, while the Y-axis indicates that between the word and negative SSV. We use diamonds to denote positive word lists, and crosses for negative word lists. Ideally, we want the two opposite groups of data to separate from the hyperplane (intersecting x-y plane on y=x) as far as possible. With adjectives being the group where most candidate words are present in the training sets, we can clearly see the data points dispersing and showing the tendency to cluster into groups. Nevertheless, we also see a lot of negative words being falsely classified as the positive group, which could be explained with the way that a synset of a word in WordNet is constructed. Although most words in the corpus are explicate through direct explanation, some of the words are described through negation of positive words. For instance, the word “impolite” has the synset of “not polite”. Unfortunately, negating words such as “not” in this example are considered as stop words that will be removed during the formulation of gloss vectors. In the end, the gloss vectors could deliver the exact opposite polarities after the removal stop words. As for noun and verb, two groups of data points can be distinguished in the charts. On the contrary, adverb possesses data points that mainly stay on the boundary that should presumably separate the two polarities.

We take a closer look into the three most frequent components in the polarity vectors as shown in TABLE II. Although we are conducting the 5-fold validation, the first fold holdout is randomly selected as our example. For both adjective and noun, only one common word is present in the top-three list of both vectors (Adjective: Mark, Noun: Act). Adverb, on the other hand, has the exact same words for both the positive and negative vectors (Manner, Degree, Extreme). We therefore suggest that when a candidate word is being compared with both vectors, these common components make it difficult for the word vector to be distinguished from either of the two, hence producing erroneous results. As for verb, although it has two common components in the top-three lists of both vectors, the training sets for verb have a greater amount of data than those of adverb, so the impact of common terms could be balanced from the extra gloss components generated from the more extensive data source.

In the end, we want to look back at the comparison result inspired by Patwardhan [8] listed in TABLE I. Technically, their method was only used to classify the semantic relatedness of nouns featured in the study of Miller and Charles [25] and Rubensteirn and Goodenough [26], so a fair comparison could not be conducted against their results. Still, it is worthwhile to point out how the same concept can produce different performances when applied to different parts-of-speech and used with different corpus. The gloss vectors of the word “abundant” generated by both methods are displayed in Figure 8 to demonstrate the difference between the numbers of components in the vectors. All of the components were stemmed and lemmatized to their most basic terms. For the two-level vectors, we only listed some of the components that exist more than ten times, since it would take up more than one page to show the full content of the vector. It is evident that the extra level of sweep in the glossary of WordNet produced more noise than benefits, as the word “Britain” shows up more than 70 times in the vector. Moreover, quantity nouns without obvious sentiment polarities are often included in the vector as well, such as volume(11), measure(35), number(45), etc. These words outnumber some of the true emotion-prone terms such as great(17) to a great degree, preventing the algorithm to make a legitimate classification.

We also attempted to perform the classification through pairwise comparison of words. Initially, we tried to perform semantic analysis through the comparison of the gloss vector of the candidate words with that of the words “good” and “bad”, assuming that if the candidates were closer to either of the words, then their glosses (after removing the stop word and lemmatized) would contain more common keywords. The problem with this approach is that when using the two-level gloss vectors, the noise would hinder any practical classification, whereas using the one-level gloss vector would not have enough components to do a meaningful vector calculation.

### TABLE II. MOST FREQUENT DIMENSIONS IN THE VECTORS.

<table>
<thead>
<tr>
<th>Part-of-speech</th>
<th>Positive</th>
<th>Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adjective</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mark</td>
<td>59</td>
<td>Lack</td>
</tr>
<tr>
<td>Show</td>
<td>51</td>
<td>Mark</td>
</tr>
<tr>
<td>Character</td>
<td>47</td>
<td>Cause</td>
</tr>
<tr>
<td>Adverb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Manner</td>
<td>147</td>
<td>Manner</td>
</tr>
<tr>
<td>Degree</td>
<td>22</td>
<td>Degree</td>
</tr>
<tr>
<td>Extreme</td>
<td>18</td>
<td>Extreme</td>
</tr>
<tr>
<td>Noun</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Quality</td>
<td>68</td>
<td>Act</td>
</tr>
<tr>
<td>Act</td>
<td>59</td>
<td>Person</td>
</tr>
<tr>
<td>Feel</td>
<td>57</td>
<td>State</td>
</tr>
<tr>
<td>Verb</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Make</td>
<td>67</td>
<td>Make</td>
</tr>
<tr>
<td>Give</td>
<td>33</td>
<td>Move</td>
</tr>
<tr>
<td>Feel</td>
<td>20</td>
<td>Feel</td>
</tr>
</tbody>
</table>

**1-level Gloss Vector:**
- Abundant: present, 1; great, 1; quantity, 1;

**2-level Gloss Vector:**
- Abundant: great, 17; quantity, 17; exist, 11; time, 35; person, 30; body, 11; substance, 12; organ, 12; acid, 13; extent, 25; degree, 36; people, 10; number, 45; physical, 23; system, 11; state, 38; pass, 10; dai, 20; quality, 21; character, 12; measure, 35; express, 22; volume, 11; blood, 10; cell, 12; form, 21; gener, 12; amount, 21; act, 19; unit, 47; make, 24; pair, 10; large, 28; produce, 10; ancient, 12; northern, 15; property, 10; show, 14; work, 11; perform, 10; item, 13; denot, 15; britain, 73; ......

![Fig. 8. Comparison between amount of components of 1-level and 2-level gloss vectors of the word abundant.](image-url)
Consequently, instead of using only one indication word as the polarity identifier (good / bad), we take the whole list of keywords as the training set to test the unknown candidates in the testing set. By doing this, we are forming a diverse data set in the vector space while at the same time preserving the clarity of the vectors.

V. CONCLUSION

In our work, we demonstrated word polarity classification with the use of sentiment sensitive vector (SSV), which is based on the gloss vector derived from the glossary of WordNet. First we tried the method of Patwardhan [8] by doing a two-level sweep of the WordNet corpus to form the gloss vector for the candidate word. The gloss vector of the candidate word was then compared with those of the words “good” and “bad” to find the higher cosine similarity of the two to determine the sentiment polarity of the candidate. Afterwards, we improved our method by generating gloss vectors that use only one level of gloss extraction. SSV is then developed by assembling gloss vectors generated from every word in a list of keywords [3] with predetermined polarity instead of using only one word as the polarity indication. Finally, we performed 5-fold cross validations on the word lists for both training and testing sets. With these adjustments, we are able to exclude the potential noise that would possibly influence the outcome. Moreover, by using multiple folds of words as the training set to form the SSV, the dimension of vector and accuracy could be greatly enhanced.

Our method has proven to be straight-forward and effective. Nevertheless, more experiments can be conducted on different data sets, since the proportions of words with different part of speeches are not similar in the current data source. We also look to implement the deep neural network (DNN) as one of our future goals.

ACKNOWLEDGMENT

We are grateful to the anonymous reviewers for their insightful comments. This research was supported by the Ministry of Science and Technology of Taiwan under grant MOST 103-3111-Y-001-027.

REFERENCES

