# Sentiment Classification for Web Search Results

Heng-Li Yang, Hung-Chang Huang

Department of Management Information Systems, National Chengchi University, Taiwan
yanh@nccu.edu.tw, 101356016@nccu.edu.tw

## Abstract

This study proposes an approach to display Google search results with different classes of sentimental orientations: (1) positive, negative, or neutral, (2) positive or negative, (3) positive or non-positive, and (4) negative or non-negative. A prototype, called as GSCS was also constructed to retrieve the search results of smartphones, tablets, and notebooks from Google. With a single click, the GSCS would help users easily get the opinions that they want to meet their different needs. For classifying documents, we suggest a two-level sentiment classification approach. At the sentence level, sentences are first classified into positive, negative, or neutral, and then the sentiment labels of the sentences were used in the classification of documents. We also demonstrated that our two-level sentiment classification (first sentence level and then document level) outperformed the document-level-only sentiment classification.

**Keywords:** Opinion mining, Sentiment analysis, Sentiment classification, Web opinions, Google search

## 1 Introduction and Research Background

The improvement of search engines, blogs, web forums, and social networks has made it easy for people to access different kinds of opinions or product reviews on the Internet. However, for users who are looking for these articles, the Google search engine does not present its search results in an appropriate way. The articles with "thumbs-up" and those with "thumbs-down" are mixed, so users have to click one by one article to check the opinion orientation. However, more often, users know what kind of opinions they want before they click the search button. For example, while looking for a restaurant that is not too bad, the negative opinions are more important than the positive ones. In other words, users may have different search requirements for different opinion orientations. Therefore, for mitigating the burden of users, it would be desirable to have a system that divides and presents the search results based on the opinions for a product.

However, to our best knowledge, though Serrano-Guerrero et al. [2] surveyed a number of web services carrying out sentiment analysis, little research has explored this aspect of search engines. Only Eirinaki et al. [3] built an opinion search engine called AskUs for English reviews on vacuums, cameras, and DVD players. However, their study had two weaknesses. First, their web opinions were only classified as positive vs. negative; however, sometimes users hope to read three classes (i.e., positive, negative, neural) or two other classes (e.g., positive vs. non-positive). Second, their sentiment classification was based on the document scores, which was computed by considering only the orientation of opinion words in the body and title. Such computation needs more refinement to considering the body structure and other features.

The main goal of this research is to propose a system based on Google, called Google sentiment classification system (GSCS). It would classify the Chinese search results of a product queried by a user into several divisions labeled either in three classes (positive, negative, or neutral), or two classes (e.g. negative vs. non- negative) as the user requests. In addition, we would consider the whole document structure and other features for classifying sentiment.

Opinion mining, or sentiment analysis, refers to a technique or a research topic that analyzes the opinions, sentiments, evaluations, or attitudes of humans towards the target entity from the given text [1]. The target entity could be a product, service, organization, person, event, issue, topic, or attribute. This process is highly related to data mining, text mining, information retrieval, and machine learning [1, 4-5]. Previous studies have conducted sentiment analyses on many fields, e.g., film comments, product reviews, or tweets [6-9].

An opinion contains five main elements: holder, target entity, entity aspect, time, and sentiment orientation [1]. The target entity and the entity aspect are also called the opinion target. If we assume that there is only one opinion target in each document, that is, all words and sentences are opinions for the same product, we would have no chance of discovering opinions for other products, and this might lead to a misclassification. Opinion target identification is a

*Corresponding Author: Heng-Li Yang; E-mail: yanh@nccu.edu.tw
DOI: 10.3966/160792642019122007003

method that tries to find the opinion targets in each sentence in a document. Researchers have developed several methods for opinion target identification, and most of the methods for Chinese are rule-based. Lin and Chao [10] applied both rule-based methods and machine learning methods to identify tourist attractions, the opinion target in their case, from Chinese blogs. They also found that bloggers often use co-references such as partial names, abbreviations, or other special terms to refer to the same tourist attraction. Thus, collecting these co-references is necessary. Lu [11] introduced a simple heuristic rule-based method to identify opinion targets in Chinese news. Ma and Wan [12] searched candidate opinion targets in news comments by using the centering theory, and took the most possible one as the opinion target after evaluating the candidates.

There are several levels to consider while conducting opinion mining: document, sentence, clause, phrase, and word levels [1]. In this research, an experiment was performed at both sentence level and document level. At the sentence level, we discuss opinion target identification and sentiment classification. At the document level, we only discuss the latter.

In this study, an experiment was conducted on web reviews of portable devices, including smartphones, tablets, and laptops for demonstrating the precision and flexibilities of the system. A GSCS prototype was also built. In the following sections, taking the above dataset as examples, we describe our approach to GSCS construction.

# 2 Proposed Approach to Construct Google Sentiment Classification System

## 2.1 Document Collection and Processing

At the very beginning, a product domain should be chosen. In this research, for constructing the prototype, we chose the domain of portable devices, including smartphones, tablets, and notebooks. Next, we collected the product names, nouns, or terms used for referring to these products. These names were categorized into brand names, series names, model names, and informal names. For example, "How much does everyone expect that HTC Desire 816 should be?" Here, "HTC" is a brand name, "Desire" is a series name of certain HTC products, and "816" is the model name of a product from the "Desire" series. That is, a brand has one or more series, and a series has one or more models.

However, in Chinese, informal names are more commonly used than formal names on the Internet. For example, "哀鳳 (Ai-Feng)" means "iPhone" and "XZ" means "Xperia Z". They are co-references and are important for opinion target identification. However, their derivations are less structured, so these terms are collected through manual works.

The purpose of our prototype is only for demonstrating the feasibility of our approach. Thus, we limit the possible keywords submitted to Google. There are three types of products in our domain: smartphones, tablets, and notebooks. We selected six hot brands for each type of product, one series for each brand, and one model for each series. As a result, there were 54 keywords, but some of the keywords duplicated; for example, "Apple" is a vendor that sells all the three products. After removing duplicate keywords, we had 48 keywords. We used these keywords as queries to Google, searching for the documents in two famous forums of mobile devices in Taiwan, namely Mobile01 and PTT. We retrieved the top 10 results for each search. Some of the searches did not retrieve a sufficient number of results, and some of the results were invalid, such as a page-not-found error. Finally, we collected 934 valid documents in our dataset. In addition, for the sentence-level analysis, these documents were parsed into 33425 sentences.

## 2.2 Opinion Target Identification

In a web review, the author may not always comment on only one product but uses examples or comparisons in order to point out the differences from other products. While the opinion target changes, the sentiment orientation with respect to the product changes too. A positive description about a product may be a negative description about another product. Therefore, identification of the opinion target is necessary for our research. We used the CKIP parser (http://parser.iis.sinica.edu.tw/) from Academia Sinica to obtain sentence parsing trees and then developed a rule-based opinion target identification method for this research by combining the rules of Lu [11], and Ma and Wan [12]. These rules were applied to the extraction of sentence-level features.

The rules refer to two terms, namely opinion target candidates and the search target. *Opinion target candidates* are defined as brand names, series names, model names, or informal names that are observed in a sentence. The *search target* is defined as the above name, the keyword for submitting to Google. We developed seven ordinal rules as the following. These rules are applied to assign opinion target candidate in the descending order.

Rule 1. Take the search target if there is a term representing an equivalent relation (e.g., "相同 (same)" or "一樣 (as … as …)") and the search target appears in the sentence.

Rule 2. Take the opinion target candidate following an advocate verb.

Rule 3. Take the opinion target candidate at the head of the sentence.

Rule 4. Exclude the opinion target candidates

following a preposition or a verb (non-advocate verb).

Rule 5. Take the last opinion target candidate.

Rule 6. Take the identified opinion target from the previous sentences, three sentences at the most.

Rule 7. Take the search target as the opinion target.

Rules 1-5 are for sentences containing the opinion target candidates, and the rest are for sentences not containing the opinion target candidates. The last rule assigns the search target as the default opinion target, which is same as the assumption made in document-level opinion mining.

## 2.3 Manual Tagging

Both Mobile01 and PTT do not have a scoring or rating function for their product reviews that can be transformed into sentiment labels. Thus, the sentiment labels are obtained by manual tagging, that is, assigning each document and each sentence sentiment labels (tags) manually. At the document level, all of our 934 documents were tagged. Each document was given a sentiment label for positive, negative, or neutral. We had 273 positive, 259 negative, and 402 neutral documents. At the sentence level, there were 33425 sentences, resulting in a considerable amount of effort for tagging them all. Instead, we selected several representative samples for training a sentence-level classifier. The best classifier would be then used for predicting all of the 33425 sentences. We grouped the documents into three types of products (smartphones, tablets, and laptops), three types of search targets (brand, series, and model), two web forums (Mobile01 and PTT), and three sentiment labels (positive, negative, and neutral), resulting in 54 groups of documents. For each group, four or five documents were selected for sentence-level human tagging. A total of 227 documents with 11012 sentences were selected. We had 1601 positive, 1083 negative, and 8328 neutral sentences. Three persons tagged each sample document or sentence independently. In the first run, about 80% sentences were judged in the same sentiment orientation. The rest conflicts were further discussed one by one in the second run for deciding the sentiment label.

## 2.4 Sentiment Lexicon

### 2.4.1 Lexicon of Opinion Words

The source of sentiment lexicon contains NTU sentiment dictionary (NTUSD) and HowNet-VSA. NTUSD [13] is a Chinese sentiment lexicon, containing 9365 positive terms and 11230 negative terms. HowNet-VSA is the Chinese/English vocabulary for sentiment analysis of HowNet developed by Dong (http://www.keenage.com). It provides six categories of terms, both in English and in Chinese. The Chinese part includes 9193 terms, which contain 4566 positive terms, 4370 negative terms, 219 degree terms, and 38 advocate verbs[1].

Our steps to build the lexicon of opinion words are as follows:

(1) Compare NTUSD and HowNet-VSA.

(a) In HowNet-VSA, some terms are recorded as both positive terms and negative terms. They were added to the initial neutral term set $O_{init}$, which required further work to specify the sentiment orientation of the terms.

(b) Terms appearing only in either NTUSD or HowNet-VSA were added to the initial positive term set $P_{init}$ or the initial negative term set $N_{init}$ according to their original sentiment orientation.

(c) Terms in both NTUSD and HowNet-VSA and having the same sentiment orientation were added to the initial positive term set $P_{init}$ or the initial negative term set $N_{init}$ according to their original sentiment orientation.

(d) Terms in both NTUSD and HowNet-VSA but with different sentiment orientations were added to the initial neutral term set $O_{init}$.

(2) Intersect $P_{init}$, $N_{init}$, and $O_{init}$ with the parsed corpus to filter terms not present in the corpus, called $P_{filter}$, $N_{filter}$, and $O_{filter}$.

Furthermore, some terms in a particular domain may have different meanings from their general uses in Chinese. For example:

*e.g., "筆電產品的本質就是**輕薄**好攜帶加上優異的效能表現。"*
*("The essence of a laptop is **tiny**, easy to take, and good performance.")*

*e.g., "這男的總是在用言語**輕薄**和調戲別人。"*
*("This man always **looks down on** and teases others in words.")*

The word "輕薄" in the former example means "thin, light, or tiny" in Chinese, while that in the latter means "look down on somebody." The phenomenon still holds even in the same domain. For example:

*e.g., "這支手機的電池可以用很久。"*
*("The battery of this phone lasts **long**.")*
*e.g., "這支手機的瀏覽器讀網頁讀很久。"*
*("It takes a **long** time for the browser of this phone to read pages.")*

[1] In fact, HowNet has 836 positive emotional terms, 3730 positive appraisal terms, 1254 negative emotional terms, and 3116 negative appraisal terms. Emotional terms are terms that express human emotions, such as "快樂 (happy)" or "傷心 (sad)," and appraisal terms are terms that express human feelings about something, such as "簡單 (easy)" or "困難 (hard)." In this study, we were only concerned with their orientation. Since they are in Simplified-Chinese, they would be first translated into Traditional-Chinese in this study.

The word "久 (long)" in the two cases is used for describing time, and both cases are in the domain of smartphones. However, it is obvious that the former is positive and the latter is negative. These cases are related to aspect-level opinion mining. When the same term is used for describing different product features, they may present different sentiments [14]. In a general view, the effect does not only exist between opinion words and product features but also between opinion words and the other words. If an opinion word presents some specific sentiment when it co-occurs with another term, this pair of terms is called collocation.

Church and Hanks [15] introduced pointwise mutual information (*PMI*), a method for computing the independence of two words, $w_1$ and $w_2$.

$$PMI(w_1, w_2) = \log_2 \frac{P(w_1, w_2)}{P(w_1)P(w_2)} \quad \textbf{(1)}$$

Here, P($w_1$, $w_2$) denotes the probability that $w_1$ and $w_2$ both occur, P($w_1$) represents the probability that w1 occurs, and P($w_2$) indicates the probability that $w_2$ occurs. When *PMI* is zero, the two words are independent, which means that an occurrence of one word does not affect the occurrence of the other. A large *PMI* means that the two words are much likely to co-occur. Conversely, a small *PMI* means that the two words are less likely to co-occur. Thus, *PMI* is an appropriate way for measuring and extracting the collocations for a specific domain [16].

Based on *PMI*, Turney [17] introduced the concept of semantic orientation (*SO*), which computes the sentiment of a word *w*, whose sentiment is still unknown.

$$SO(w) = PMI(w, w^+) - PMI(w, w^-) \quad \textbf{(2)}$$

Here, $w^+$ and $w^-$ denote a known positive and a known negative opinion word, respectively. Turney [17] used "excellent" and "poor," for example. A positive *SO* means that *w* is positive in terms of the sentiment, while a negative *SO* means that *w* is negative in terms of the sentiment. Therefore, *SO* can be used for defining the sentiment of opinion words. In our research, we changed the calculation of *PMI* in *SO* because we had already obtained the sentiment labels of the sentences by manual tagging. The *PMI* was then changed for computing the independence of opinion words and positive/negative sentences, called $PMI^+$ / $PMI^-$, respectively.

3. Compute the $PMI^+$ $PMI^-$, and *SO* of each term in $P_{filter}$, $N_{filter}$, and $O_{filter}$.

Positive words with significantly negative *SO*s should be redefined as negative words, and negative words with significantly positive *SO*s should be redefined as positive words. However, *SO* is not always positively correlated to sentiment. Positive opinion words whose $PMI^+$ is less than zero, which means that they are not positively correlated to positive sentences, should not be categorized as positive opinion words. In the same way, negative opinion words whose $PMI^-$ is more than zero should not be categorized as negative opinion words.

4. Redefine the opinion words.

(1) Terms whose $PMI^+ > 0$ and $SO > 0$ in $P_{filter}$ or $O_{filter}$ are added to the final positive opinion word set $P_{opn}$.

(2) Terms whose $PMI^- > 0$ and $SO < 0$ in $N_{filter}$ or $O_{filter}$ are added to the final negative opinion word set $N_{opn}$.

(3) After the standardization of *SO* in $N_{filter}$ by z-score, terms whose $PMI^+ > 0$ and $SO > 1$ are added to the final positive opinion word set $P_{opn}$.

(4) After the standardization of *SO* in $P_{filter}$ by z-score, terms whose $PMI^- > 0$ and $SO < -1$ are added to the final negative opinion word set $N_{opn}$.

Finally, we have 947 positive opinion words and 482 negative opinion words.

### 2.4.2 Lexicon of Collocations

The following shows how to build a lexicon of collocations:

(1) Remove words that are not related to collocations by using the part-of-speech (POS) tags. We reserved words tagged as "A (adjective)," "Na (nouns)," "Nv (nominalized verbs)," "FW (foreign words)," and verbs other than "V_2 ('have')" and "SHI (to-be verbs)."

(2) For each word in the lexicon of opinion words $a_i$, look for all the words that co-occur with $b_i$, named $B_i$. Each word $b_{ij}$ in $B_i$ is a word that might be a collocation with $a_i$. The set of all the pairs ($a_i$, $b_{ij}$) is called $C_{init}$.

(3) Compute the *PMI* of each pair ($a_i$, $b_{ij}$) in $C_{init}$.

(4) Remove the pairs in $C_{init}$ whose $PMI \leqq 0$, called $C_{filter}$, which has 13022 collocations.

(5) Compute the $PMI^+$, $PMI^-$, and *SO* of each pair in $C_{filter}$.

A collocation ($a_i$, $b_{ij}$) is formed by an opinion word $a_i$ and the other word $b_{ij}$. For instance, "方便 (convenient)" is a positive opinion word, and "功能 (function)" is a word that collocates with "方便", which might mean that some of the functions of the device are convenient. In this case, irrespective of whether "功能" appears or not, the sentiment of the sentence or the document stays unchanged. Therefore, we remove the collocations whose sentiment is the same as its opinion word.

(6) Remove the collocations whose sentiment is consistent with its opinion word.

(a) Collocations whose $PMI^+ > 0$, $SO > 0$, and opinion words were negative were added to the positive collocation set $P_{col}$.

(b) Collocations whose $PMI^- > 0$, $SO > 0$, and opinion words were positive were added to the negative collocation set $N_{col}$.

Finally, we obtained 441 positive collocations and 1670 negative collocations.

### 2.4.3 Lexicon of Degree Adverbs and Negation Adverbs

In addition to opinion words and collocations, words such as negation adverbs and degree adverbs, which may affect opinion words and collocations, should also be considered. Many studies have pointed out that negation adverbs are important factors [1, 3, 18]. Negation adverbs can reverse the sentiment easily [18], such as "好 (good)" and "不好 (not good)". Degree adverbs are also important factors affecting sentiment [19]. Degree adverbs do not change the sentiment if there is only one opinion word, but if there are multiple opinion words, degree adverbs help to balance the sentiment. For example:

*e.g., "我最近買的新手機很好，只可惜貴了一點。"*
*("The new phone I bought recently is very nice, but it's just a little expensive.")*

Here, the degree of positive sentiment of "很好 (very nice)" is more than the degree of negative sentiment of "貴了一點 (a little expensive)", so the sentiment of this sentence should be positive. If degree adverbs were not considered, this sentence would be neutral because it has one positive and one negative opinion word.

The negation adverbs and degree adverbs are first taken from Chang [20], which categorizes degree adverbs into three levels: strong, medium, and weak. HowNet-VSA also provides degree adverbs at six levels, which are "極其 (extremely) / 最 (most)", "很 (very)", "較 (more)", "稍 (a few)", "欠 (less)", and "超 (over)". We rearranged these six levels into three levels. "極其 (extremely) / 最 (most)" and "超 (over)" are strong degrees, "很 (very)" and "較 (more)" are medium degrees, and "稍 (a few)" and "欠 (less)" are weak degrees. Finally, we integrated the adverbs from the two sources. The summary statistics are listed in Table 1.

**Table 1.** Statistics of negation adverbs and degree adverbs

| Chang [20] | Strong | | Medium | | Weak | | Negation |
|---|---|---|---|---|---|---|---|
| number of words | 20 | | 59 | | 38 | | 56 |
| HowNet-VSA | Extremely/Most | Over | Very | More | A few | Less | (N/A) |
| number of words | 69 | 30 | 42 | 37 | 29 | 12 | 0 |
| Integrated the two sources | Strong | | Medium | | Weak | | Negation |
| number of words | 107 | | 115 | | 54 | | 56 |

### 2.4.4 Other Lexicons

Some words are used for sentence-level feature extraction and opinion target identification, which contain the advocate verbs, words for expressing equivalent relations, and words for expressing conclusions. Advocate verbs are taken from HowNet-VSA. Words for expressing equivalent relations and words for expressing conclusions are taken from E-HowNet[2] by keyword searching. The keywords for the former are "相等 (equal)" and "一樣 (same)", and the keywords for the latter are "結論 (conclude)" and "總而言之 (in summary)". Thus, we obtain 35 advocate verbs, 29 words for expressing equivalent relations, and 21 words for expressing conclusions.

## 2.5 Sentence-Level Features

At the sentence level, we extracted 10 features from our raw data, the 11012 sentences, for the classification experiment.

(1) *fs11 (proportion of positive opinion words), fs12 (proportion of negative opinion words), fs21 (proportion of positive collocations), and fs22 (proportion of negative collocations):* It is a common to consider all opinion words and all collocations to be features. However, there are few opinion words or collocations in each sentence, with respect to the number of all opinion words and all collocations. It would result in a quite sparse and large feature matrix, which goes against the machine learning for classifiers. Thus, we attempted to summarize the sentiment opinion words and collocations. For opinion words, we computed the proportions of positive and negative opinion words as two features. For instance:

*e.g., s1 "這台筆電既好用又便宜，但是已經賣光了。"*
*("This notebook is good to use and cheap, but it had already sold out.")*

There are three opinion words in s1, which contains two positive words "好用 (good to use)" and "便宜 (cheap)", and one negative word "可惜 (pity)". In this case, fs11 is 2/3 or 0.67 and fs12 is 1/3 or 0.33.

2 http://ehownet.iis.sinica.edu.tw/

*e.g., s2 "這台筆電的外型超漂亮的，但價格有點貴就是了。"*
*("The notebook looks so beautiful, but its price is a little expensive.")*

There are two collocations in s2, which contains a positive collocation "讚 (beautiful); 外型 (look)", and a negative collocation "貴 (expensive); 價格 (price)". In this case, fs21 is 1/2 or 0.5 and fs22 is 1/2 or 0.5.

(2) *fs31 (modified proportion of positive opinion words) and fs32 (modified proportion of negative opinion words):* We assigned weights to the adverbs related to opinion words. The weight for negation adverbs is -1, which means an inversion of sentiment. Chang [22] tuned several combinations of weights of degree adverbs in Chinese and found the best combinations to be 3, 2, and 1.5 for strong, medium, and weak degrees, respectively.

*e.g., s3 "我最近買的新手機很不錯，只可惜貴了一點。"*
*("The new phone I bought recently is very nice, but it's just a little expensive.")*

"不錯 (nice)" is a positive opinion word modified by a medium-degree adverb "很 (very)", and "貴 (expensive)" is a negative opinion word modified by a weak-degree adverb "一點 (a little)". In this case, fs31 is 2*1/2 = 1 and fs32 is 1.5*1/2 = 0.75.

In addition, a degree adverb can be modified by a negation adverb, and a negation adverb can be modified by a degree adverb. They have different meanings, such as "不是很好 (not very good)" and "很不好 (very 'not' good)". The "很 (very)" in the former is modified by "不是 (not)", resulting in a less degree of strength but still a positive sentiment. On the other hand, the "不 (not)" is modified by "很 (very)" in the latter, resulting in a medium degree but a negative sentiment. If we simply multiply the weights, we will get 1.5*(-1), which is the same as in the latter case. To solve the former case, we adjust the degree level to its weaker level and remove the negation weight if the degree adverb is modified by a negation adverb; in other words, 3, 2, and 1.5 are adjusted to 2, 1.5, and 1 for strong, medium, and weak, respectively. For instance:

*e.g., s4 "這台平板雖然不貴，但用起來也不是很順。"*
*("Though the tablet is not expensive, it does not work very fluently either.")*

"貴 (expensive)" is a negative opinion word modified by a negation adverb "不 (not)", and "順 (fluently)" is a positive opinion word modified by a negation adverb "不是 (not)". In this case, fs31 is 2*1/2 = 1 and fs32 is 1*(-1)/2 = -0.5.

(3) *fs41 and fs42 (relation between opinion target and search target):* We have mentioned that the sentiment changes with the opinion target. Here, we discuss the relation between the opinion target and the search target in the brand–series–model structure. When we are looking for the opinions of a brand, the opinions of its series and models are also included. Based on the relations, we developed the following rules that generate two features, spread in [0, 1] as a pair to express the distance between the opinion target (*OT*) and the search target (*ST*). (1, 1) means that they are an exact match, and (0, 0) means that they are a complete mismatch.

(a) If *ST* is a model name, then
- If *OT* and *ST* are the same model, (fs41, fs42) = (1, 1).
- If *OT* and *ST* are different models of the same series, (fs41, fs42) = (0.67, 1).
- If *OT* and *ST* are different series of the same brand, (fs41, fs42) = (0.33, 1).
- If *OT* and *ST* are different brands, (fs41, fs42) = (0, 0).

(b) If *ST* is a series name, then
- If *OT* and *ST* are the same series, (fs41, fs42) = (1, 1).
- If *OT* and *ST* are different series of the same brand, (fs41, fs42) = (0.5, 1).
- If *OT* and *ST* are different brands, (fs41, fs42) = (0, 0).

(c) If search target is a brand name, then
- If *OT* and *ST* are the same brand, (fs41, fs42) = (1, 1).
- If *OT* and *ST* are different brands, (fs41, fs42) = (0, 0).

(4) *fs43 (number of sentences that searches previous sentences for opinion target):* If there is no available opinion target candidate in a sentence, its previous sentences may have an opinion target. Although it may identify the right opinion target, there is still a chance of identifying the wrong one. The higher the number of previous sentences that it searches, the greater is the possibility of a wrong identification. Therefore, we count the number of sentences that it searches above.

(5) *fs50 (interrogative sentence):* In web forums, many users may ask questions about the product or ask for help on operations. Many of these sentences contain negative words. For instance:

*e.g., "Padfone S 開 4G 分享會很耗電嗎？"*
*("Does Padfone S consume a lot of energy when sharing its 4G network?")*

However, this sentence should be neutral rather than negative for Padfone *S* because it does not mean that Padfone *S* really consumes a lot of energy but is asking about the energy consumption condition of Padfone *S*.

Thus, whether a sentence is interrogative or not is an important feature for classifying neutral sentences. This feature is binary, that is, *0* for non-interrogative and *1* for interrogative.

### 2.6 Document-Level Features

We extract the following 15 features for training document-level classifiers.

(1) *fd10 (average sentiment of sentences):* To summarize the sentiment of the sentences in a document as the sentiment of the document, an average of sentiment scores is a simple way. We sum up the sentiments of the sentences, in terms of 1 for positive, -1 for negative, and 0 for neutral, in a document except the title, and then divide the value by the number of sentences. We do not compute the majority because neutral sentences are usually the most in a document.

(2) *fd21 (proportion of positive sentences), fd22 (proportion of negative sentences), fd23 (proportion of neutral sentences), fd24 (proportion of non-negative sentences), and fd25 (proportion of non-positive sentences):* Further, the proportion of each sentiment is another way to express the sentiment of a document. Here, we compute the proportions of five sentiments, including positive, negative, neutral, non-negative, and non-positive.

(3) *fd31 (sentiment of title):* The influence of a sentence changes with its position. The title is a critical factor. In [3], the score of the title is 10 times larger than the score of the content. Here, we take the sentiment of the title as a single feature.

(4) *fd32 (average sentiment of head section), fd33 (average sentiment of middle section), and fd34 (average sentiment of bottom section):* In formal writing, each section of an article has its use. This writing skill is called "起承轉合" in Chinese, which partially matches the structure of introduction, elucidation of the theme, and conclusion in English. From this point of view, the first section and the last section have more importance. Although in web forums, informal writings are much more than formal writing, they still, briefly, follow the structure. So, we divided the document almost equivalently into three sections, called the head, middle, and bottom sections and compute the average sentiment of these sections.

(5) *fd41 (2/1/2 weighted average sentiment of head/middle/bottom sections) and fd42 (1/2/3 weighted average sentiment of head/middle/bottom sections):* We also tried to compute the weighted average of the three sections. Consulting two experts in Chinese writing, we have two points of view. One is to assign the same weight for the head section and the bottom section, which is two times larger than that assigned to the middle section. Thus, we assigned 2, 1, and 2 to the three sections, respectively. The other is to assign a larger weight to the later sections. We assigned 1, 2, and 3 to the three sections, respectively.

(6) *fd35 (sentiment of conclusion):* Sometimes, we use words such as "總而言之 (in summary)", "因此 (therefore)", or "結論 (in conclusion)" to tell readers the conclusion of the article. A conclusion is located at the end of an article or in the bottom section. Here, we further discuss the uses of words in the bottom section. For example:

*e.g., "總而言之，買這支手機是正確的選擇。"*
*("In summary, it's a right choice to buy this phone.")*

If the above sentence appears at the end of an article, the article is most probably positive, and there is no need to consider other sentences. This is important and effective for our classification. We have collected the words used for expressing conclusions from E-HowNet. The sentiment of the last sentence that contains these words is extracted as a feature.

(7) *fd51 (relative number of words), fd52 (relative number of sentences), and fd53 (average number of words per sentence):* The longer an article is, the more the sentiment that it may contain. The structure of an article is expressed by its statistics, such as the number of words or sentences. The relative number of words/sentences is the number of words/sentences in an article divided by the average number of words/sentences of all articles, and the average number of words per sentence is the number of words divided by the number of sentences in an article.

## 3 Experiment Results

### 3.1 Sentence-Level Sentiment Classification

At the sentence level, we only trained the positive/negative/neutral classifier. We selected supervised machine learning methods that are commonly used in sentiment classification, namely the support vector machine (SVM), naïve Bayes classifier (NB), and k-nearest neighbor (KNN) algorithm. As a simple and readable method, the J48 decision tree algorithm was also used. The tools that we used were LibSVM [21] for SVM and Weka 3.62 for NB and KNN.

All of the 11012 tagged sentences were the training data, so we evaluated the results by using a 20-fold cross validation. With different parameters of the above methods, we run 37 experiments at the sentence level. Each experiment was run five times iteratively, and the average performance was obtained. At the sentence level, we did not focus on any certain class, so we chose the method that had the highest average F-measure. The best one was the J48 decision tree, with an average F-measure of 0.77.

### 3.2 Document-Level Sentiment Classification

#### 3.2.1 Four Types of Classifiers

The following four types of classifications were

performed at the document level:

(1) Positive, negative, or neutral (T1)
(2) Positive or negative (T2)
(3) Positive or non-positive (T3)
(4) Negative or non-negative (T4)

The experimental design was the same as that for the sentence-level classification except the criteria for choosing the best method. For T1 and T2 classifications, the importance of each class was equivalent, so we chose the method with the highest average F-measure first. When there were multiple methods with the same F-measure, the one with the highest accuracy was chosen. On the other hand, for T3 and T4 classifications, the positive class in T3 and the negative class in T4 were more important, so we evaluated the method based on the criteria for the positive/negative class, respectively. The method with the highest average F-measure was chosen; when there were multiple methods with the same F-measure, the one with the highest recall was chosen.

The experimental results including the best methods and their performance for the four classifications are presented in Table 2.

**Table 2.** Experimental results of document level

| Classifier | Best Method | Performance | | | |
|---|---|---|---|---|---|
| | | Avg. F-measure | Avg. Precision | Avg. Recall | Accuracy |
| T1 | LibSVM, rbf kernel | 0.77 | 0.79 | 0.77 | 0.77 |
| T2 | LibSVM, rbf kernel | 0.87 | 0.89 | 0.88 | 0.88 |
| T3 | KNN | 0.81 | 0.81 | 0.80 | 0.84 |
| T4 | KNN | 0.83 | 0.84 | 0.88 | 0.79 |

#### 3.2.2 Feature Selection

Not all of the features were of a high quality for the classification. Thus, we could remove features that had no use or little use with respect to efficiency improvement. We ranked the features by using SVMAttributeEval module in Weka, removing the lowest ranked features one by one, and evaluated the results. We kept removing features until the performance decreased by more than 0.01. In the case of T1, fd31, fd32, fd25, fd22, and fd35 were reserved. In the case of T2, fd10, fd31, fd35, fd52, and fd22 were reserved. In the case of T3, no features were removed. In the case of T4, fd51, fd41, and fd52 were removed.

#### 3.2.3 Comparisons

Although our system is designed for Chinese sentiment classification, we still compared it with Eirinaki et al. [3], which was for English. In their research, they only conducted the binary classification (positive or negative), and their system was evaluated in terms of accuracy. Therefore, we compared our T2 classifier with their system. The accuracy of our T2 classifier was 0.87, which was equivalent to that of their classifier for DVD players. The GSCS still provided other flexibilities of the T1, T3, and T4 classifiers, but there are no studies available for comparison.

Further, most of previous studies conducted at the document level directly extracted the features for opinion words and collocations. Thus, based on our dataset, we also made a comparison, and observed that the two-level approach was significantly better ($p < 0.05$) than the document-level-only approach for all classifications, T1, T2, T3 and T4 (their F-Scores were in the range of 0.45 to 0.71).

## 4 System Prototyping

The GSCS prototype was built after the classification experiment.

### 4.1 System Structure and Environment

The system structure is illustrated in Figure 1. The left side shows the external units, including the user, programs, or web services. The middle is the system logic units. The right side are databases. The units are connected by arrows, which also indicate the flow of data.

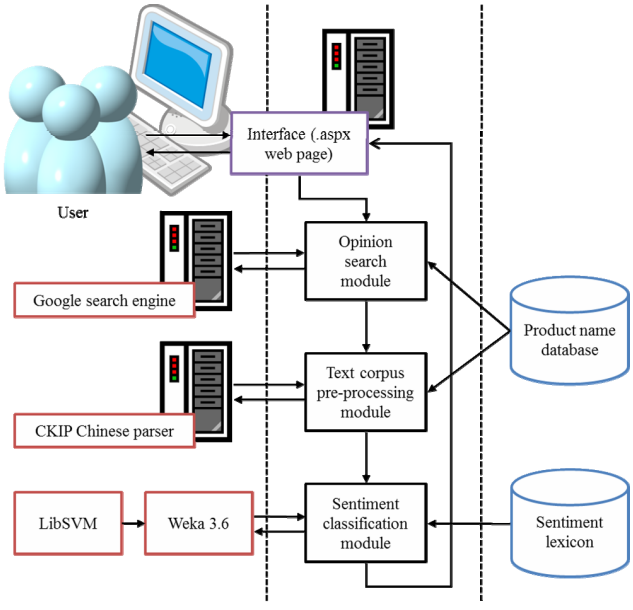


**Figure 1.** System prototype structure

Four external web services and programs were used. The Google search engine acted as the source of search results. CKIP of Sinica was used for preprocessing the documents. Weka and LibSVM supported the sentiment classification.

There were three modules coded by us to act as the system logic units. The opinion search module connects to the Google search engine and retrieves the search results of product names from users by sending URLs with query strings. In addition to avoiding users from inputting unrelated keywords, the module would match the input to our name database. The functions of pre-processing module include the extraction of titles and contents from the source code, identification of product names, and the parsing conducted using the API of CKIP. The last and the most important is the sentiment classification module, which includes feature extraction and classification at sentence and document levels. The J48 decision tree at the sentence level is implemented in the module, which reduces the time for I/O, while the document-level classification uses Weka and LibSVM.

For databases, since the classifiers were trained in our experiments, the corpus is not included in the implementation. The GSCS works with two databases. One is the product name database, and the other is the sentiment lexicon.

The prototype was developed on Windows and presented on ASP.NET web pages, and the programming language was C#.

## 4.2 System Operation and User Interface

The operation procedure for the GSCS is as follows:

(1) The user inputs the keyword that he/she wants to search.

(2) The system requests for search results on PTT and Mobile01 for the keyword by sending a URL with the query strings to the Google search engine, and then, retrieve the articles from the top 10 results.

(3) Conduct preprocessing, including the extraction of titles and contents and the identification of product names.

(4) Parse all the documents by using the API of CKIP.

(5) For each document, first, extract the sentence-level features and classify the sentences in the document, and then, extract the document-level features.

(6) The system classifies all the documents by using Weka, for each classification from T1 to T4.

(7) The classification results are read and returned to the user interface.

(8) The user can select between the four classification methods, and the results are processed at the client side. Figure 2 shows an example of the presentation of T1 with the keyword “Samsung.” The documents with positive, neutral, and negative orientations are shown in the left, middle, and right sections, respectively.


**Figure 2.** Presentation of T1 classification

## 4.3 Efficiency

To evaluate the system efficiency, we randomly selected 10 keywords from the portable device names, and recorded the processing time in each process. On average, each complete run took 186 seconds. The longest took 310 seconds, while the shortest one took 95 seconds. We found that parsing wasted most of the time. It took 88% of the total time to parse the documents. However, parsing was almost a necessary step in Chinese language processing, so our system still depended on the CKIP system. The process that took the second highest amount of time was the sentence-level feature extraction and classification because of the matching process between sentences and words.

# 5 Conclusions and Future Research

This study proposes an approach to display Google search results with different classes of sentimental orientations, which consists of three major functions: opinion search, document processing, and sentiment classification. A prototype, GSCS, was also constructed. With a single click, the GSCS would help

users easily get the opinions that they want to meet their different needs.

For classifying documents, we suggest a two-level sentiment classification approach. At the sentence level, sentences are first classified into positive, negative, or neutral, and then the sentiment labels of the sentences were used in the classification of documents. This study has the flexibility to provide four types of classifiers. In addition, the experimental result shows that the two-level method is significantly better than document-level-only classification, so our system design can improve its classification effectiveness. A feature selection at the document level is also tested. The proportion of negative sentences, sentiment of the title, and sentiment of the conclusion were found to be the most important features.

This study stands for an example of sentiment-wise search result diversification, which provides a new direction for future exploration. It is especially important and useful for product search. As the practice implications, the GSCS could benefit both personal users and businesses. Personal users can search for the opinions they need and clearly know which opinions are worth reading, resulting in decision-making improvement. Businesses can easily identify negative comments on their services or products and find possible solutions early.

There are still some limitations in our research. One is text preprocessing, and the other is aspect-based opinion mining. In web texts, some symbols, phrases, and slogans might have special meanings if used in particular situations. Furthermore, they often contain punctuation that may cause errors while parsing Chinese texts. In text analysis studies, they are recognized as noise and thus removed. As a result, we may lose the implicit sentiment information in these Internet slangs. Text normalization not only helps parsing but also helps sentiment analysis [6]. If first pre-processed by a Chinese Internet slang "translator" [22], which helps normalize web texts into formal texts, the sentimental classification of our GSCS may be improved.

We also found that some of the misclassification in our study might be attributed to user preferences for product features. Future studies may mine the sentiments of different product features to help system users with specific preferences for obtaining more relevant sentiment information. In addition, this study build our prototype based on Google. However, our prototype should work for Bing or Yahoo! Search or other information retrieval engines. Future research may try other engines.

## Acknowledgments

The authors would like to thank the Ministry of Science and Technology, Taiwan, for financially supporting this research under Contract No. NSC 101-2410-H-004-015-MY3 and MOST 107-2410-H-004-097-MY3.

## References

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

[2] J. Serrano-Guerrero, J. A. Olivas, F. P. Romero, E. Herrera-Viedma, Sentiment Analysis: A Review and Comparative Analysis of Web Services, *Information Sciences*, Vol. 311, pp. 18-38, August, 2015.

[3] M. Eirinaki, S. Pisal, J. Singh, Feature-based Opinion Mining and Ranking, *Journal of Computer and System Sciences*, Vol. 78, No. 4, pp. 1175-1184, July, 2012.

[4] H. Chen, M. Sun, C. Tu, Y. Lin, Z. Liu, Neural Sentiment Classification with User and Product Attention, *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, Austin, TX, November, 2016, pp. 1650-1659.

[5] X. Liao, X. Xu, J. Pan, G. Chen, Detect Online Review Spammers based on Comprehensive Trustiness Propagation Model, *Journal of Internet Technology*, Vol. 18, No. 3, pp. 637-644, May, 2017.

[6] S. Ahmed, A. Danti, Effective Sentimental Analysis and Opinion Mining of Web Reviews Using Rule Based Classifiers, in: H. Behera, D. Mohapatra (Eds.), *Computational Intelligence in Data Mining—Volume 1. Advances in Intelligent Systems and Computing*, Vol. 410, Springer, 2016, pp. 171-179.

[7] R. S. Jagdale, V. S. Shirsat, S. N. Deshmukh, Sentiment Analysis on Product Reviews Using Machine Learning Techniques, in: P. Mallick, V. Balas, A. Bhoi, A. Zobaa (Eds.), *Cognitive Informatics and Soft Computing. Advances in Intelligent Systems and Computing*, Vol 768, Springer, 2019, pp. 639-647.

[8] K. Ravi, V. Ravi, A Survey on Opinion Mining and Sentiment Analysis: Tasks, Approaches and Applications, *Knowledge-Based System*, Vo. 89, pp. 14-46, November, 2015.

[9] M. Malik, S. Habib, P. Agarwal, A Novel Approach to Web-based Review Analysis Using Opinion Mining, *Procedia Computer Science*, Vol. 132, pp. 1202-1209, 2018.

[10] C. J. Lin, P. H. Chao, Tourism-Related Opinion Detection and Tourist-Attraction Target Identification, *International Journal of Computational Linguistics and Chinese Language Processing*, Vol. 15, No. 1, pp. 37-60, March, 2010.

[11] B. Lu, Identifying Opinion Holders and Targets with Dependency Parser in Chinese News Texts, *Proceedings of the Human Language Technologies: 2010 Annual Conference of the North American Chapter of the ACL (NAACL HLT),* Los Angeles, CA,USA, 2010, pp. 46-51.

[12] T. Ma, X. Wan, Opinion Target Extraction in Chinese News Comments, *Proceedings of the International Conference on Computational Linguistics (COLING) Poster Volume*, Beijing, China, 2010, pp. 782-790.

[13] L. W. Ku, H. H. Chen, Mining Opinions from the Web:

Beyond Relevance Retrieval, *Journal of American Society for Information Science and Technology*, Vol. 58, No. 12, pp. 1838-1850, October, 2007.

[14] X. Ding, B. Liu, P. S. Yu, A Holistic Lexicon-based Approach to Opinion Mining, *Proceedings of the Conference on Web Search and Web Data Mining (WSDM)*, Stanford, California, 2008, pp. 231-240.

[15] K. W. Church, P. Hanks, Word Association Norms, Mutual Information, and Lexicography, *Proceedings of the 27th Annual Conference of the ACL*, Vancouver, British Columbia, Canada, 1989, pp. 76-83.

[16] A. M. Popescu, O. Etzioni, Extracting Product Features and Opinions from Reviews, *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing (EMNLP)*, Vancouver, Canada, 2005, pp. 339-346.

[17] P. D. Turney, Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews, *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, PA, USA, 2002, pp. 417-424.

[18] B. Liu, Sentiment Analysis and Subjectivity, in: N. Indurkhya, F. J. Damerau (Eds), *Handbook of Natural Language Processing,* 2nd ed., Taylor and Francis Group, Boca, 2010, pp. 627-666.

[19] J. Liu, S. Seneff, Review Sentiment Scoring via a Parse-and-Paraphrase Paradigm, *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore, Singapore, 2009, pp. 161-169.

[20] Y. J. Chang, *A Study on Library Users' Satisfaction Evaluation Using Sentiment Analysis*, Master Thesis, National Chung Hsing University, Taichung, Taiwan, 2012.

[21] C. C. Chang, C. J. Lin, LIBSVM: A Library for Support Vector Machines, *ACM Transactions on Intelligent Systems and Technology*, Vol. 2, No. 3, pp. 1-27, April, 2011.

[22] H. L. Yang, H. C. Huang, Q. F. Lin, An Internet Slang Translator Based on Decision Tree and Bigram Language Model. *Journal of e-Business*, Vol. 7, No. 1, pp. 25-48, March, 2014.

## Biographies

**Heng-Li Yang** is a professor in the Department of Management Information Systems, National Cheng-Chi University, Taiwan. His research interests include virtual community, data & knowledge engineering, etc.

**Hung-Chang Huang** received M.S. degree from Department of Management Information Systems, National Cheng-Chi University, Taiwan.