

# Extracting Resource Terms for Sentiment Analysis

**Lei Zhang**

Department of Computer Science  
University of Illinois at Chicago  
851 S. Morgan St, Chicago IL 60607  
lzhang3@cs.uic.edu

**Bing Liu**

Department of Computer Science  
University of Illinois at Chicago  
851 S. Morgan St, Chicago IL 60607  
liub@cs.uic.edu

## Abstract

Existing research on sentiment analysis mainly uses sentiment words and phrases to determine sentiments expressed in documents and sentences. Techniques have also been developed to find such words and phrases using dictionaries and domain corpora. However, there are still other types of words and phrases that do not bear sentiments on their own, but when they appear in some particular contexts, they imply positive or negative opinions. One class of such words or phrases is those that express resources such as water, electricity, gas, etc. For example, “*this washer uses a lot of electricity*” is negative but “*this washer uses little water*” is positive. Extracting such resource words and phrases are important for sentiment analysis. This paper formulates the problem based on a bipartite graph and proposes a novel iterative algorithm to solve the problem. Experimental results using diverse real-life sentiment corpora show good results.

## 1 Introduction

Sentiment analysis or opinion mining has been an active research area in recent years (e.g., Pang and Lee 2008; Turney, 2002; Wiebe *et al.* 2004; Hu and Liu, 2004; Kim and Eduard, 2004; Wilson *et al.* 2005; Popescu and Etzioni, 2005; Riloff *et al.* 2006; Esuli and Fabrizio, 2006; Mei *et al.* 2007; Stoyanov and Cardie, 2008). Researchers have studied the problem at the document level, sentence level and aspect level to determine the sentiment polarity expressed in a document, in a sentence and on an aspect of an entity (see the surveys (Pang and Lee, 2008) and (Liu, 2010)). One type of key information used in almost all existing sentiment analysis techniques is a list of sentiment words (or opinion words). Positive sentiment words are words ex-

pressing desired states or qualities, e.g., *good*, *amazing*, and *excellent*, and negative sentiment words are words expressing undesirable states or qualities, e.g., *bad*, *crappy*, and *ugly*.

A key characteristic of these words is that they themselves bear sentiments. They are frequently used in sentiment analysis tasks. However, it is also important to recognize that sentiment analysis based only on these words (or phrases) is far from sufficient. There are still many other types of expressions that do not bear sentiments on their own, but when they appear in some particular contexts, they imply sentiments. In (Liu, 2010), several such expressions and their corresponding opinion/sentiment rules are introduced. We believe that all these expressions have to be extracted and associated problems solved before sentiment analysis can achieve the next level of accuracy. One such type of expressions involves resources, which occur frequently in many application domains. For example, *money* is a resource in probably every domain (“*this phone costs a lot of money*”), *gas* is a resource in the car domain, and *ink* is a resource in the printer domain. If a device consumes a large quantity of resource, it is undesirable. If a device consumes little resource, it is desirable. For example, the sentences, “*This laptop needs a lot of battery power*” and “*This car uses a lot of gas*” imply negative sentiments on the laptop and the car. Here, “gas” and “battery power” are resources, and we call these words *resource terms* (which cover both *words* and *phrases*).

In terms of sentiments involving resources, the rules in Figure 1 are applicable (Liu, 2010). Rules 1 and 3 represent normal sentences that involve resources and imply sentiments, while rules 2 and 4 represent comparative sentences that involve resources and also imply sentiments, e.g., “*this washer uses much less water than my*

1.	Positive	←	consume no or little resource
2.			consume less resource
3.	Negative	←	consume a large quantity of resource
4.			consume more resource

**Figure 1:** Sentiment polarity of statements involving resources.

old GE washer”. To the best of our knowledge, there is no reported algorithm that extracts resource terms. In this paper, we propose an iterative algorithm to extract them from a domain corpus, e.g., a set of product reviews. In the above example sentence, we want to extract “water” as a resource term.

The most related work to ours is the product aspect/feature extraction (e.g., Hu and Liu, 2004, Popescu and Etzioni, 2005, Kobayashi *et al.* 2007, Scaffidi *et al.* 2007, Titov and McDonald, 2008, Stoyanov and Cardie. 2008, Wong *et al.*, 2008, Zhao *et al.*, 2010). A resource in a domain is often an aspect or implies an aspect. For example, in “this camera uses a lot of battery power”, “battery power” clearly indicates *battery life*, which is an aspect of the camera entity. However, there are some important differences between resources and other types of aspects. The key difference is that resource terms often contribute directly to sentiments (e.g., based on the quantity that is consumed), while other aspects may not. e.g., “picture quality” in “the picture quality of this camera is great,” where “great” solely determines the sentiment of the sentence. Thus, resource terms require special treatments in sentiment analysis. In this paper, we focus on identifying and extracting resource terms.

This paper models the extraction problem with a bipartite graph and proposes a novel circular definition to reflect a special reinforcement relationship between *resource usage verbs* (e.g., *consume*) and *resources* (e.g., *water*) for resource extraction. We call the proposed method MRE (*Mutual Reinforcement based on Expected values*). Based on the definition, the problem is solved using an iterative algorithm. To initialize the iterative computation, some global *seed resources* are employed to find and to score some strong resource usage verbs. These scores are applied as initialization for the iterative computation in the bipartite graph for any application domain. When the algorithm converges, we obtain a ranked list of candidate resource terms. Our experimental results based on 7 real-life data sets show the effectiveness of the proposed method. It outperforms 5 strong baselines.

## 2 Related work

As we discussed in the introduction, this work is mainly related to product aspect extraction. Hu and Liu (2004) proposed a technique based on association rule mining to extract frequent nouns and noun phrases as product aspects. They also introduced the idea of using sentiment words to find additional (infrequent) aspects. Popescu and Etzioni (2005) improved the precision of this method by determining whether a noun/noun phrase is indeed a product aspect by computing the pointwise mutual information (PMI) score between the phrase and class discriminators, e.g., “xx has”, “xx comes with”, etc., where xx is a product class word, and using Web search.

A dependency based method is proposed in (Zhuang *et al.*, 2006) to extract aspects for a movie review application. Dependency relations are also used in (Qiu *et al.* 2011) to extract both aspects and sentiment words. Zhang *et al.* (2010) augmented this method by introducing aspect ranking. Wang and Wang (2008) proposed a similar bootstrapping method but not based on dependencies. In (Kobayashi *et al.* 2007), a pattern mining method was proposed to find extraction patterns. Statistics from the corpus are employed to determine the extraction confidence.

Other works on aspect extraction use topic modeling and probabilistic modeling to capture and group aspects at the same time (e.g., Mei *et al.*, 2007; Titov and McDonald, 2008; Lu *et al.* 2009; Zhao *et al.*, 2010; Wang *et al.* 2010; Jo and Oh, 2011). In (Su *et al.*, 2008), a clustering method was also proposed with mutual reinforcement to identify aspects.

However, all these existing works focused on extracting aspects in general. They do not specifically identify resource terms, which are a special type of aspects, and need additional techniques to recognize them.

Our work is also related to the general information extraction problem. There are two main approaches to information extraction: rule-based and statistical. Early extraction systems are mainly based on rules (e.g., Riloff, 1993). In statistical methods, the most popular models are Hidden Markov Models (HMM) (Rabiner, 1989;

Jin *et al.*, 2009), and Conditional Random Fields (CRF) (Lafferty *et al.*, 2001). CRF has been used in extracting aspects and topics (e.g., Stoyanov *et al.*, 2008, Jakob and Gurevych, 2010). However, a limitation of CRF is that it only captures local patterns rather than long range patterns. Also, CRF is a supervised method, but our method is a bootstrapping method which needs no supervision but only a few initial global resource seeds.

Our proposed method is also related to the Web page ranking algorithm HITS (Kleinberg, 1999), which finds hub and authority pages based on the hyperlink structure of the Web pages. However, our method is quite different as we have a different formulation. We will discuss the details in Section 3. HITS is also one of the baseline methods that will be compared with the proposed MRE technique in the evaluation section. Our method outperforms HITS considerably.

### 3 The Proposed Method

In this section, we present the proposed technique. Let us use the following two example sentences to develop the idea and the algorithm:

1. *This car uses a lot of gas.*
2. *This car uses less gas than Honda Civic.*

We call the first sentence a *normal sentence*, and the second sentence a *comparative sentence*.

From these two sentences, we can make the following observation:

**Observation:** The sentiment expressed in a sentence about resource usage is often determined by the triple,

(verb, quantifier, noun\_term),

where *noun\_term* is a noun or a noun phrase

In the first sentence, “uses” is the main verb, “a lot of” is a quantifier phrase, and “gas” is a noun representing a resource. In the second sentence, “uses” is also the main verb, “less” is a comparative quantifier, and “gas” is again a resource as a noun. We want to use such triples to help identify resources in a domain.

We notice that using only a pair,

(verb, noun\_term), or  
(quantifier, noun\_term)

is not sufficient. The pair (verb, noun\_term) is unsafe because such pairs are very common since subject-verb-object (SVO) is the most common English sentence structure, and the object is usually a noun term. Using (quantifier,

noun\_term) is also unsafe as the meaning of the noun terms following quantifiers can be diverse.

By no means do we say that any above triple implies the last noun term is a resource. For example, “colors” is not a resource in “*this car got many colors*”. The triples only find candidate resources, which need to be further analyzed (see Section 3.2).

Since it is unsafe to use the pair (verb, noun\_term) or (quantifier, noun\_term), we use only triples for candidate resource extraction. Due to the fact that it is easy to compile the main expressions of quantifiers, we just need to extract verbs and noun terms to discover candidate resources which are the noun terms. The quantifiers that we use in this work are listed in Table 1.

Quantifiers
some, several, numerous, many, much, more, most, less, least
a large/huge/small/tiny number of
a large/huge/small/tiny quantity/amount of
lot/lots/tons/ton/plenty/deal/load/loads of
[a] few/little

**Table 1:** A list of quantifiers

#### 3.1 Extract Triples and Build a Graph

Since our algorithm is based on triples, we now discuss how to extract them. To extract triples from a corpus, part-of-speech (POS) tagging is first performed on each sentence. Verbs and nouns are then identified based on their POS tags. Verbs are words tagged as VB, VBD, VBZ, VBG, VBN, and VBP. Nouns are words tagged as NN and NNS. In addition, we regard a phrase with continuous POS tags of NN and NNS as a noun phrase, e.g., “spray/NN gel/NN” is seen as a single noun phrase “spray gel”. In English grammar, quantifiers usually precede and modify noun terms. Thus, after locating a quantifier in a sentence, we extract its associated noun term, which directly follows the quantifier. After obtaining the noun term, we further exploit the dependency relation to find the associated verb in the sentence, since there is an assumed *verb-object* relationship between the verb and the noun. The relationship can be determined by a dependency parser. In our work, we approximate the dependency by making use of a text window in the sentence. It works quite well. Thus we did not use a dependency parser, which tends to be inefficient. We choose the closest verb in a text window (e.g., 10 words) before the noun as the

verb part of the triple. Note that verbs such as “is”, “was”, “am”, “are”, “were”, “have”, “has”, and “had” are not used since they usually do not express resource usages. Finally, we lemmatize both the verb and the noun and store them only in the lemmatized format in a triple.

With all extracted triples, we build a bipartite graph based on the verb set  $V$ , the noun set  $N$ , and the set of links  $L$  between  $V$  and  $N$ . A link  $(i, j)$  is in  $L$  if there is a triple involving a verb  $i \in V$  and a noun term  $j \in N$ . Note that in this graph, we do not use quantifiers, which are only used to identify candidate verbs and nouns.

### 3.2 The Proposed Algorithm

We now present the proposed algorithm, which relies on the bipartite graph to encode a special kind of mutual enforcement relationship between resource usage verbs and resource terms. Before diving into the details of the algorithm, we define the following concepts.

**Definition (Resource Term):** A resource term represents a physical or virtual entity that can be consumed or obtained in order to benefit from it.

Some resources are general, which exist in many different application domains, i.e., “money” in “*this TV costs me a lot of money*”. Other resources are more domain-specific, e.g., “onboard memory” in “*the phone uses more onboard memory*”.

**Definition (Resource Usage Verb):** A resource usage verb (or *resource verb* for short) is a verb that can express resource usage.

Likewise, some resource verbs are general and can modify many different resource terms, e.g., “uses” in “*this car uses much more gas*”, “*this washer uses a lot of water*”, and “*this program uses a lot of memory*.” Many others are more resource-specific, and tend to frequently co-occur with specific resources, e.g., “*spent*” in “*I spent too much money to buy the car*”.

It seems that we can solve the problem of extracting resource terms using a simple graph propagation strategy. That is, given an application domain corpus, the user first provides a few seed resource terms. Using the bipartite graph, we can identify some resource verbs by following the links of the graph. The newly identified resource verbs are then used to identify new resource terms. The process continues until no more resource terms or verbs can be found.

However, this simple strategy has some major

problems. First, as many resource verbs and terms are domain-specific, asking the user to provide some seeds for each domain is non-trivial. Second, many nouns (or verbs) in the triples may not be resources (or resource usage verbs), e.g., “*this car comes with many colors*.” Any error resulted in the propagation can generate more errors subsequently.

With these concerns in mind, we propose a more sophisticated iterative algorithm. To solve the first problem above, we take a global approach. Instead of asking the user to provide some seed resources for each domain, we simply provide some global resource seeds, e.g., *water*, *money*, and *electricity*. Then in each application, the user does not need to do anything. Using these global resource seeds, we want to identify some good resource usage verbs. These verbs act as the initialization for the discovery of additional resource terms in each domain based on the domain corpus. The proposed method thus consists of two main stages. The first stage is only done once and the results are used for individual application domains as the initialization.

#### Stage 1: Identifying Global Resource Verbs

Global resource verbs are those verbs that can express resource usage of many different resources, e.g., *use* and *consume*. We can use a bipartite graph constructed from a large data set to find them. The following observations help us formulate the solution:

1. A global resource verb has links to many different resource terms. The more diverse the resource terms that a verb can modify, the more likely it is a good global resource verb.
2. Conversely, the more global resource verbs a resource term is associated with, the more likely it is a genuine resource term.

These two observations indicate that the global resource verbs and the resource terms have a mutual enforcement relationship, which can be modeled by the Web page ranking algorithm HITS exactly. We give a brief introduction to the HITS algorithm (Kleinberg, 1999) below.

The objective of HITS (Hyperlink-induced topic search) is to find Web pages that are authorities and hubs. A good authority page is a page pointed to by many pages, and a good hub is a page that points to many pages. There is a mutual reinforcement relationship between authority pages and hub pages.

Given a set of Web pages  $S$ , HITS computes an *authority score* and a *hub score* for each page

in  $S$ . Let the number of pages to be studied be  $n$ . We use  $G = (S, E)$  to denote the (directed) link graph of  $S$ , where  $E$  is the set of directed edges (or links) among the pages in  $S$ . We use  $M$  to denote the adjacency matrix of the graph.

$$M_{ij} = \begin{cases} 1 & \text{if } (i, j) \in E \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

Let the authority score of page  $i$  be  $A(i)$ , and the hub score of page  $i$  be  $H(i)$ . The mutual reinforcing relationship in HITS is defined as follows:

$$A(i) = \sum_{(j,i) \in E} H(j) \quad (2)$$

$$H(i) = \sum_{(i,j) \in E} A(j) \quad (3)$$

We can write them in a matrix form. We use  $\mathbf{A}$  to denote the column vector with all authority scores, and use  $\mathbf{H}$  to denote the column vector with all hub scores:

$$\mathbf{A} = M^T \mathbf{H} \quad (4)$$

$$\mathbf{H} = M \mathbf{A} \quad (5)$$

To solve the equations, the widely used method is power iteration, which starts with some random values for the vectors, e.g.,  $\mathbf{A}^0 = \mathbf{H}^0 = (1, 1, \dots, 1)^T$ . It then continues to compute iteratively till convergence. Note that the initial values do not generally affect the final ranking of authorities and hubs.

In our scenario, global resource verbs act as hubs and resource terms act as authorities. We provided a list of common resources (seeds) (see Section 4). Using these seeds, we extract triples from the corpus and produce a link graph as discussed in Section 3.1. The noun term set  $N$  consists of only these seed resource terms, and the  $V$  set consists of only those verbs which form triples with the  $N$  set. HITS is then applied on the graph. After HITS converges, each candidate resource verb has a hub score. We normalize them to the 0-1 interval. The resulting values are used to initialize the system for discovering resource terms from each application domain. That is, we do not need to execute stage 1 anymore.

## Stage 2: Discovering Resource Terms in a Domain Corpus

Given the global resource verb values from stage 1 and a domain corpus, the stage 2 system identifies resource terms from the domain corpus.

In this stage, we still start with a bipartite graph as in the first stage. The graph can be con-

structed as discussed in Section 3.1 by extracting triples from the domain corpus. On one side of the bipartite graph, it is the set of candidate resource terms  $N$  (noun terms) and on the other side, it is the set of candidate resource (usage) verbs  $V$ . For each  $i \in V$ , we want to compute its likelihood of being a resource verb, denoted by  $u(i)$ , and for each noun term  $j \in N$ , we want to compute its likelihood of being a resource term, denoted by  $r(j)$ . If  $i$  and  $j$  are in a triple, a link  $(i, j)$  is in the link set  $L$ .

An obvious question is: Can we use HITS here as in stage 1? The answer is no. Unlike stage 1, the  $N$  set here is no longer a set of true resources, but only a list of noun terms, which are just candidate resources. A verb modifying multiple noun terms does not necessarily indicate that the verb is a resource usage verb. For example, it could be a general verb like “get”. Also, as mentioned earlier, it is not always the case that if a noun term is modified by many verbs, it is a resource term. For example, it could be a topic word like “car” for the car domain. Applying the simple reinforcement relation in HITS is ineffective as we will see in the experiment section. To introduce the proposed technique, we make the following observations:

1. If a noun term is *frequently* associated with a verb (including quantifiers), the noun term is more likely to be a genuine resource term.
2. If a verb is *frequently* associated with a noun term (including quantifiers), it is more likely to be a genuine resource verb.

These two observations indicate that we should take verb and noun term co-occurrence frequency into consideration, which cannot be used in HITS. To consider frequency, we turn the frequency into a probability and make use of the expected value to compute scores for the verbs and noun terms, rather than summation in HITS.

In probability, given a random variable  $X$ , its *expected value* is defined as

$$E[X] = \sum_i p_i x_i \quad (6)$$

where  $x_i$  is a possible outcome of the random variable  $X$  and  $p_i$  is the probability of  $x_i$ .

For our case, we have the following definitions for  $u(i)$  and  $r(j)$ .

$$u(i) = \sum_{(i,j) \in L} p_{ji} r(j) \quad (7)$$

$$r(j) = \sum_{(i,j) \in L} p_{ij} u(i) \quad (8)$$

where

$$p_{ij} = \frac{c(i, j)}{\sum_{(k, j) \in L} c(k, j)} \text{ and}$$

$$p_{ji} = \frac{c(i, j)}{\sum_{(i, k) \in L} c(i, k)}$$

$c(i, j)$  is the frequency count of the link  $(i, j)$  in our corpus.  $p_{ij}$  is thus the probability of link  $(i, j)$  among all links from different verbs  $i$  to a noun  $j$ .  $p_{ji}$  is the probability of link  $(i, j)$  among all links from different nouns  $j$  to a verb  $i$ . We called this proposed algorithm MRE (*Mutual Reinforcement based on Expected values*)

### Smoothing the Probabilities

Although the idea is reasonable, we found an important issue when computing expected values. If a noun term  $j$  occurs only once, and it is connected with a strong resource verb  $i$ , its ranking value becomes very high. Due to its low frequency, the expected value of  $r(j)$  is just the value of  $u(i)$ . In many cases, the value may be even higher than some frequent noun terms, whose value may be reduced by being associated with some non-resource verbs. This situation is not desirable. Since for sentiment analysis application, we should rank those frequent resource terms at the top instead of the terms which only occur once in the corpus.

The problem is that the probabilities of verbs or nouns are not reliable due to limited data. In order to handle infrequent verbs or noun terms, we smooth the probabilities to avoid probabilities of 0 or 1. The standard way of doing this is to augment the count of each distinctive verb/term with a small quantity  $\lambda$  ( $0 \leq \lambda \leq 1$ ) or a fraction of a verb or noun term in both the numerator and denominator. Thus any verb and noun term will have a smoothed probability as follows.

$$p_{ij} = \frac{\lambda + c(i, j)}{\lambda |N| + \sum_{(k, j) \in L} N(k, j)} \quad (9)$$

$$p_{ji} = \frac{\lambda + c(i, j)}{\lambda |V| + \sum_{(i, k) \in L} N(i, k)} \quad (10)$$

This is called the *Lidstone smoothing* (Lidstone's law of succession) (Lidstone, 1920). We use  $\lambda$  to 0.01, which performs well. In the equations,  $|V|$  is the total number of verbs and  $|N|$  is the total

### Algorithm: MRE ( $Q, G$ )

**Input:** A global resource verb set  $Q$  with their hub scores computed from HITS in stage 1, and  $G$  is the bipartite graph

**Output:** a ranked list of candidate resource terms

1.  $u^0(i) \leftarrow H(i)$  of verb  $i$ , if *verb*  $i \in Q$
2.  $u^0(i) \leftarrow \arg \min_{r \in Q} \{H(r)\}$ , if *verb*  $i \notin Q$
3. **Repeat** till convergence
4.  $r^{n+1}(j) = \sum_{(i, j) \in L} p_{ij} u^n(i)$
5.  $u^{n+1}(i) = \sum_{(i, j) \in L} p_{ji} r^n(j)$
6. normalize  $r(j)$  and  $u(i)$
7. Output the ranked candidate resource terms based on their  $r(j)$  score values.

**Figure 2:** The proposed MRE algorithm

number of noun terms in the graph.

Note that with smoothing, the original bipartite graph becomes a complete bipartite graph. Each added link is given a very small probability as computed using Equations (9) and (10).

### The Computation Algorithm

The computation algorithm for the proposed method MRE is given in Figure 2.  $Q$  is the set of verbs from stage 1, and  $G$  is the bipartite graph. To initialize the iterative computation, we assign the hub score from stage 1 to each verb  $i \in V$  as its initial score  $u^0(i)$  if  $i$  is in  $Q$  (line 1). If  $i$  is not in  $Q$ ,  $u^0(i)$  is given the minimum value of the hub scores of all verbs in  $Q$  (line 2).

After this initialization, the algorithm proceeds iteratively until convergence. We will describe the convergence characteristic of the algorithm in Section 4.5.

Finally, we note that unlike HITS, which converges to the same hub and authority (steady-state) scores regardless the initialization. For MRE, the initialization makes a big difference as we will see in the evaluation section.

## 4 Evaluation

We now evaluate the proposed MRE method. We first describe the data sets, evaluation metrics, and then the experimental results. We also compare MRE with 5 baseline methods.

### 4.1 Data Sets and Global Resource Seeds

We used seven (7) diverse data sets to evaluate our technique. These data sets were crawled from the Web. Table 2 shows the domains (based on their names) and the number of sentences in each

Data sets	Car	Washer	Paint	Printer	Haircare	Mobile	TV
# of Sent.	56880	9997	1655	16314	29347	25354	23901

**Table 2.** Experimental data sets

Data sets	Car	Washer	Paint	Printer	Haircare	Mobile	TV	Ave.
TF	0.40	0.20	0.60	0.80	0.40	0.40	0.20	0.43
TFR	0.40	0.40	0.40	0.80	0.40	0.40	0.60	0.49
HITS	0.60	0.40	0.20	0.80	0.60	0.40	0.40	0.49
MRE-NI	0.20	0.80	0.20	0.60	0.60	0.60	0.80	0.54
MRE-NS	0.60	0.60	0.60	0.80	0.60	0.40	0.40	0.57
MRE	1.00	0.80	0.60	0.80	0.60	0.80	0.80	0.77

**Table 3.** Experimental results: Precision@5

Data sets	Car	Washer	Paint	Printer	Haircare	Mobile	TV	Ave.
TF	0.40	0.20	0.70	0.60	0.30	0.50	0.50	0.46
TFR	0.30	0.50	0.60	0.50	0.40	0.40	0.50	0.46
HITS	0.50	0.60	0.50	0.70	0.50	0.50	0.40	0.53
MRE-NI	0.30	0.80	0.40	0.40	0.30	0.70	0.60	0.50
MRE-NS	0.70	0.60	0.70	0.60	0.60	0.70	0.40	0.61
MRE	0.90	0.80	0.80	0.60	0.70	0.80	0.60	0.74

**Table 4.** Experimental results: Precision@10

Data sets	Car	Washer	Paint	Printer	Haircare	Mobile	TV	Ave.
TF	0.40	0.30			0.20	0.35	0.35	0.32
TFR	0.30	0.50			0.30	0.20	0.40	0.34
HITS	0.55	0.65			0.50	0.50	0.35	0.51
MRE-NI	0.30	0.70			0.45	0.50	0.45	0.48
MRE-NS	0.60	0.65			0.50	0.55	0.45	0.55
MRE	0.75	0.70			0.65	0.60	0.55	0.65

**Table 5.** Experimental results: Precision@20

data set (“Sent.” means the sentence). Each data set contains a mixture of reviews, blogs, and forum discussions about one type of product. We split each posting into sentences and the sentences are POS-tagged using the Brill’s tagger (Brill, 1995). The tagged sentences are the input to our system MRE.

The global resource terms (resource seeds) used in the first stage of our method are: “gas”, “water”, “electricity”, “money”, “ink”, “shampoo”, “detergent”, “room” “fabric softener”, and “soap”. In stage 1 of our algorithm, we used the combined data set of those in Table 2 to compute the hub scores for global resources usage verbs found to be associated with the resource seeds through some quantifiers.

#### 4.2 Evaluation Metrics

We adopt the rank precision, also called **precision@N** metric for the experimental evaluation. It gives the percentage of correct resource terms (precision) at different rank positions. This is a popular method used in search ranking evaluation because one does not know all the relevant pages. This is also the case in our work as we do

not know how many resource terms have been mentioned in each of the data set.

#### 4.3 Baseline Methods

TF (Triple Frequency): This method finds all triples of the form (verb, quantifier, noun\_term), and then ranks them according to their frequency counts. This basically corresponds to the methods used in (Hu and Liu 2004; Popescu and Oren, 2005; Zhuang *et al.* 2006; Qiu *et al.* 2011) as it combines the frequency and dependency patterns of the triples. This method is reasonable because many triples are indeed resource usage descriptions, and those more frequent ones (ranked high) are more likely to be genuine ones.

TFR (Triple Frequency Ratio): This method is similar to the above method but it divides TF by the number of pairs (verb, noun\_term) with the same verb and the same noun term as in the triple. The reason for doing so is that such pairs are very common because **subject-verb-object (SVO)** is the most common English sentence structure, and object is usually a noun term. If the ratio of the occurrences of

the triple is small, it may not be a resource usage description and then should be ranked low because sentences containing resources are usually talking about resource usages.

**HITS:** This method simply runs the HITS algorithm in the second stage for each data set. In this case, the global initialization is not useful as HITS will reach a steady state regardless of the initialization.

**MRE-NI:** Our MRE method without initialization by the global resource usage verbs.

**MRE-NS:** Our MRE method without the probability smoothing.

#### 4.4 Results and Discussions

Tables 3-5 give the precision results for top 5, top 10, and top 20 ranked candidate resource terms. Each value in the last column gives the average precision for the corresponding row. We note that in Table 5, there are no results for “Paint” and “Printer” because no resources were found by any algorithm beyond top 10 as there are not many resources in these domains. It is also important to note that those resources that have been used as global seeds in stage 1 of our algorithm are not counted in the precision computation for the results in the tables. In other words, the discovered resource terms are all new. From the tables, we can make the following observations:

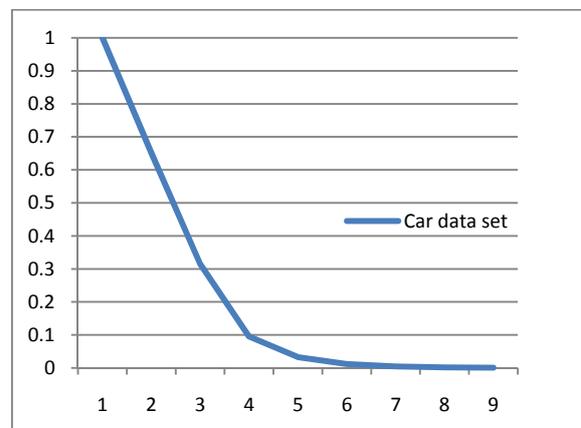
1. TF and TRF perform poorly. We believe the reason is that frequent triples or frequent triple ratio do not strongly indicate resource usages.
2. The performance of the HITS algorithm is also inferior. For only two data sets (out of 7), it performs similarly to MRE for the top 5 results. Its average results are all much worse than those of MRE.
3. Global resource verbs are very useful. As we can see, without using them (MRE-NI), the results are dramatically worse.
5. Probability smoothing also helps significantly. Without it, MRE-NS produces worse results consistently compared with MRE.
6. MRE is the best method overall. On average, it consistently outperforms every baseline method. Moreover, it does better than the 5 baseline methods on every data set at every rank position except for the data set “Printer” for the top 10 results, for which HITS is better.

From these observations, we can conclude that our proposed MRE algorithm is highly effective and it outperforms all 5 baseline methods.

#### 4.5 Algorithm Convergence

In this sub-section, we show the convergence characteristic of the proposed MRE algorithm.

Figure 3 shows the convergence behavior of MRE for the car data set, where the  $x$ -axis is the number of iterations, and the  $y$ -axis is the difference of the average 1-norm values of the vector  $r$  and vector  $u$  in two consecutive iterations. We can see that the algorithm converges quite fast, i.e., in about 8 iterations. For other data sets, they behave similarly. All of them converge within 6-9 iterations. In all experiments, the algorithm stops when the 1-norm difference is less than 0.01.



**Figure 3:** Convergent rate for car data

## 5 Conclusion

This paper proposed the problem of extracting resource words and phrases in opinion documents. They are a class of terms that are important for sentiment analysis. As we explained in the introduction section, when such resource terms appear with certain verbs and quantifiers, they often imply positive or negative sentiments or opinions. To the best of our knowledge, this work is the first attempt to discover such words and phrases. A novel iterative algorithm based on a circular definition of resource words and their corresponding verbs has been proposed. It was modeled on a bipartite graph and a special reinforcement relationship between resource usage verbs and resource terms. Experimental results based on 7 real-world opinion data sets showed that the proposed MRE method was effective. It outperformed 5 baseline methods. In our future work, we plan to improve the algorithm to make it more accurate, and also study sentiment analysis involving resource words or phrases

## References

- Brill, Eric. 1995. Transformation-Based Error-Driven Learning and Natural Language Processing: a case study in part of speech tagging. *Computational Linguistics*, 1995.
- Esuli, Andrea and Fabrizio Sebastiani 2006. Senti-WordNet: A Publicly Available Lexical Resource for Opinion Mining. *LREC 2006*
- Hu, Mingqing and Bing Liu. 2004. Mining and Summarizing Customer Reviews. *KDD 2004*
- Jakob, Niklas and Iryna Gurevych. 2010. Extracting Opinion Targets in a Single- and Cross-Domain Setting with Conditional Random Fields. In *Proceedings of EMNLP 2010*
- Jin, Wei, Hung Hay Ho, and Rohini K. Srihari. 2009. A Novel Lexicalized HMM-based Learning Framework for Web Opinion Mining. *ICML 2009*
- Jo, Yohan and Alice Oh. 2011. Aspect and Sentiment Unification Model for Online Review Analysis. In *Proceedings of WSDM 2011*
- Kim, Soo-Min and Eduard H. Hovy. 2004. Determining the sentiment of opinions. *COLING-2004*, 2004.
- Kleinberg, Jon. 1999. "Authoritative sources in hyper-linked environment" *Journal of the ACM* 46 (5): 604-632 1999
- Kobayashi, Nozomi, Kentaro Inui, and Yuji Matsumoto. 2007. Extracting Aspect-Evaluation and Aspect-of-Relations in Opinion Mining. *EMNLP*.
- Lafferty, John, Andrew McCallum and Fernando Pereira. 2001 Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proceedings of ICML*, 2001.
- Lidstone, J. George. 1920. Note on the General Case of the Bayes-Laplace Formula for Inductive or a Posteriori Probabilities. *Transactions of the Faculty of Actuaries* 1920
- Liu, Bing. 2010. Sentiment analysis and subjectivity. *Handbook of Natural Language Processing*, second edition, 2010.
- Lu, Y., C. Zhai, and N. Sundaresan. Rated aspect summarization of short comments. In *Proceedings of International Conference on World Wide Web (WWW-2009)*, 2009.
- Mei, Qiaozhu, Ling Xu, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic Sentiment Mixture: Modeling Facets and Opinions in Weblogs. In *Proceedings of WWW*, 2007.
- Pang, Bo and Lillian Lee. 2008. Opinion Mining and Sentiment Analysis. *Foundations and Trends in Information Retrieval* pp. 1-135 2008.
- Popescu, Ana-Maria and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proceedings of EMNLP*, 2005.
- Qiu, Guang, Bing Liu, Jiajun Bu and Chun Chen. 2011. Opinion Word Expansion and Target Extraction through Double Propagation. *Computational Linguistics*, Vol. 37, No. 1: 9-27.
- Rabiner, Lawrence. 1989. A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition. In *Proceedings of the IEEE*, 77(2), 1989.
- Riloff, Ellen. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of AAAI 1993*.
- Riloff, Ellen, Siddharth Patwardhan and Janyce Wiebe. 2006. Feature Subsumption for Opinion Analysis. In *Proceedings of EMNLP 2006*
- Stoyanov, Veselin and Claire Cardie. 2008. Topic Identification for Fine-grained Opinion Analysis. In *Proceedings of COLING 2008*
- Su, Qi, Xinying Xu, Honglei Guo, Zhili Guo, Xian Wu, Xiaoxun Zhang, Bin Swen and Zhong Su. 2008. Hidden Sentiment Association in Chinese Web Opinion Mining. *WWW 2008*.
- Titov, Ivan and Ryan McDonald. 2008. Modeling Online Reviews with Multi-grain Topic Models. In *Proceedings of WWW 2008*.
- Turney, Peter. 2002. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of ACL-2002*, 2002.
- Wang, Bo and Houfeng Wang. 2008. Bootstrapping both Product Features and Opinion Words from Chinese Customer Reviews with Cross-Inducing In *Proceedings of IJCNLP 2008*.
- Wang, Hongning, Yue Lu, and Chengxiang Zhai. 2010. Latent Aspect Rating Analysis on Review Text Data: A Rating Regression Approach. In *Proceedings of KDD 2010*.
- Wiebe, Janyce, T. Wilson, R. Bruce, M. Bell, and M. Martin. 2004. Learning subjective language. *Computational Linguistics*, 30(3): p. 277-308.
- Wilson, Theresa, Janyce Wiebe and Paul Hoffmann. 2005. Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *HLT/EMNLP 2005*
- Zhang, Lei, Bing Liu., Suk Hwan Lim, Eamonn O'Brien-Strain. 2010. Extracting and Ranking Product Features in Opinion Documents. In *Proceedings of COLING 2010*
- Zhao, Xin, Jing Jiang, Hongfei Yan, Xiaoming Li. 2010. Jointly Modeling Aspects and Opinions with a MaxEnt-LDA Hybrid. *EMNLP*, 2010.
- Zhuang, Li, Feng Jing, Xiao-yan Zhu. 2006. Movie Review Mining and Summarization. *CIKM 2006*.