# Analyzing Implicit Aspects and Aspect Dependent Sentiment Polarity for Aspect-based Sentiment Analysis on Informal Turkish Texts

Batuhan Kama
METU Computer Engineering Dept.
Ankara, TURKEY
batuhan.kama@ceng.metu.edu.tr

Murat Ozturk
METU Computer Engineering Dept.
Ankara, TURKEY
mozturk@ceng.metu.edu.tr

Pinar Karagoz
METU Computer Engineering Dept.
Ankara, TURKEY
karagoz@ceng.metu.edu.tr

I. Hakki Toroslu
METU Computer Engineering Dept.
Ankara, TURKEY
toroslu@ceng.metu.edu.tr

Murat Kalender
Huawei Turkey Research and Development Center
Istanbul, TURKEY
murat.kalender@huawei.com

## ABSTRACT

The web provides a suitable media for users to post comments on different topics. In most of such content, authors express different opinions on different features or aspects of the topic. In aspect based sentiment analysis, it is analyzed as to for which aspect which opinion is expressed. Once aspects are available, the next important step is to match aspects with correct sentiments. In this work, we investigate enhancements for two cases in matching step: extracting implicit aspects, and sentiment words whose polarity depends on the aspect. The techniques are applied on Turkish informal texts collected from a products forum. Experimental evaluation shows that additional steps applied for these cases improve the accuracy of aspect based sentiment analysis.

## CCS CONCEPTS

• **Information systems → Sentiment Analysis**;

## KEYWORDS

sentiment analysis, aspect, product review, implicit aspect, sentiment polarity

## 1 INTRODUCTION

Due to widespread use of the Internet and the increase in the number of mobile smart devices, terabytes of new data are posted each day. A high portion of such postings is user comments having opinionated content. In [9], sentiment analysis is defined as a research area which aims to resolve people's thoughts, attitudes and feelings on different brands, individuals and groupings. Despite the fact that document level and sentence level sentiment analysis gives an overall orientation, they do not explain what exactly people like or dislike [9]. Therefore, aspect-based sentiment analysis is introduced.

Aspect-based sentiment analysis inspects people's attitude over each specific aspect of entities. The main steps of aspect-based sentiment analysis are aspect extraction and sentiment matching, where aspect extraction step aims to find different aspects of the topic mentioned in document and sentiment matching step matches extracted sentiments to aspects to find overall sentiment over each aspect [4] [12].

In this work, we focus on the sentiment-aspect matching step. We present a basic algorithm for matching explicit aspects extracted and the explicit sentiment words. However, there are still issues remaining such as co-referencing, implicit aspects and sentiments etc. We focus on two of these issues and investigate enhancements for two cases. The first one is for the sentences in which aspects are implicit. For example, if a user comments as *The laptop is heavy* s/he actually refers to the *weight* aspect, but it is not explicitly mentioned. The second case is about sentiment words whose polarity depends on the aspect. For example, the sentiment word *high* expresses a positive opinion for the *screen resolution* of a laptop, but it a negative opinion for *price* aspect.

This work is part of a project on extracting aspect-based sentiments for products. For experimental evaluation, we have collected user comments on a specific mobile phone model from product forums in Turkish. The proposed techniques are applied on Turkish texts, however, the only language specific task is morphological analysis to determine the nouns. Experimental evaluation shows that the enhancements applied for the above mentioned two cases improve the aspect-based sentiment analysis accuracy.

This paper is organized as follows. In Section 2, related work is summarized. In Section 3, we present the basic method and proposed enhancements. Section 4 includes the experimental evaluation. We conclude the paper in Section 5 with an overview.

## 2 RELATED WORK

There are several studies on aspect based sentiment analysis in the literature [4] [12]. There are two main steps of aspect based sentiment analysis: aspect extraction [7], [11], [8] and opinion


*MEDES '17, November 7–10, 2017, Bangkok, Thailand*

ACM ISBN 978-1-4503-4895-9/17/11…$15.00
https://doi.org/10.1145/3167020.3167041

analysis in which aspect sentiment matching is performed. In this section, we summarize the previous efforts on the second step.

In the literature, there are two main approaches for aspect sentiment-matching: lexicon-based approach and supervised learning approach.

The studies under lexicon based approach have been mostly unsupervised. Sentiment lexicon, composite expressions, part-of-speech tags of sentence and rules are the commonly used features. In [3], Ding et al. propose a four-step approach. In the first step, sentiment words and phrases are marked and assigned their score from sentiment lexicon. As the second step, sentiment shifters are extracted and applied on the sentiment score. After that, *but-clauses* are handled to detect score of context-dependent sentiment words. Finally, aggregate score for each sentiment word in a document was calculated.

There are several other studies that follow lexicon based techniques. For example, Wan applied similar techniques for Chinese [13] and Blair et al. proposed a lexicon-based supervised method [2].

There are also studies that use supervised learning based techniques. Most of these work employ the similar techniques that were used in sentence-level sentiment analysis [15] [16]. In addition, a hierarchical classification approach based on sentiment ontology tree is proposed by Wei and Gulla [14]. Ganapathibhotla and Liu focused on supervised aspect-based sentiment analysis method on comparative sentences [5].

Implicit aspect extraction have been studied rarely in the literature. In [10], a link is created between an aspect and sentiment word if they co-occur in a sentence and this link becomes stronger when this two words co-occurr more often. Aspect-sentiment word pair with links whose strength above a given threshold is added to mapping list. Hai et al. propose a two-step co-occurrence association rule mining method [6]. Firstly, association rules are created by labelling frequently co-occurring sentiment words as conditions and explicit aspects as consequences. Secondly, consequences are clustered and robust rules for conditions are generated. These rules are used for discovering implicit aspect in a given new sentence.

## 3 PROBLEM DEFINITION AND PROPOSED METHOD

### 3.1 Problem Definition

Let $A$ be a set of aspects $\{a_1, ..., a_n\}$ and let $S$ be a set of sentiment terms $\{s_1, ..., s_k\}$. Given a set of terms $T = \{t_1, ..., t_r\}$ such that $T \supset AUB$, a sentence is a sequence of terms $< t_1, ..., t_q >$.

Given sequence of terms $< t_1, ..., t_q >$ that correspond to a sentence, aspect-sentiment matching is extraction of tuples $(a_i, s_j, \text{sentiment score})$, such that $s_j$ is the sentiment associated with aspect $a_i$ under the given sentiment score.

### 3.2 Proposed Method

In this section, firstly, we present our basic method for aspect sentiment matching, which extracts the explicit associations between aspect terms in the text with sentiment words. It is assumed that aspect terms are already known as a dictionary. Then the techniques for implicit aspect extraction and polarity extraction for aspect dependent sentiment words to extend the basic method are described.

*3.2.1 Explicit Aspect - Sentiment Mapping.* Explicit aspect - sentiment mapping process is represented in Figure 1 as finding noun groups, finding sentiment word groups, matching noun groups to sentiment word groups and extracting scores for aspects. The corresponding algorithm is given in Algorithm 1.

In the first step, input is a sentence, which is a sequence of terms $< t_1, ..., t_q >$ and the output is a set of *noun groups NG* $\{ng_1, ..., ng_t\}$ in this sentence. In this work, a noun group $NG$ is defined as group of nouns which are consecutive or just separated with comma or with connection word *and* (in Turkish, words *ve* and *ile* correspond to *and* in English). Note that nouns are detected through part of speech tagging. Since we work on a data set in Turkish, we used a Turkish morphological analyzer [1]. In order to find noun groups in a sentence, firstly, the position of the first noun in the sentence is detected with the help of NLP tool. Then, all consecutive nouns, commas and *ve* and *ile* words after this noun is grouped together and this new noun group is added to noun groups set. After that, the next noun in the sentence is searched in order to extract the next noun group. This process continues until no new noun is found in the sentence. The corresponding algorithm is given in Algorithm 2.

As the second step, *sentiment word groups SG*, $\{sg_1, ..., sg_r\}$ in the given sentence are extracted. Similar to the previous step, consecutive sentiment words, commas and conjunctions constitute a sentiment word group. Moreover, sentiment score of each sentiment word in sentiment word group is also kept in sentiment word group. Finally, a sentiment word group can contain sentiment enhancer (such as *very*) and sentiment shifter words (such as *not*), if there exists. If there is a sentiment enhancer in the sentiment word group, the enhancement coefficient, which is is defined as 1.5 for all sentiment enhancers in the scope of this work, is applied to increase the sentiment score of each word in the group. If any sentiment shifter is found, then the shifter coefficient, which is defined as -1 for all sentiment shifters, is applied to the sentiment word that the shifter affects. The processing for sentiment shifter and sentiment enhance are described in Algorithm 3 and Algorithm 4, respectively.

The third step is responsible for mapping extracted noun groups to extracted sentiment word groups in a sentence. At the beginning of this step, we have a set of noun groups and a set of sentiment word groups. Firstly, the number of noun groups and the number of sentiment word groups extracted are determined. All the elements of the set with smaller size must be mapped to different elements of the other set. For example, if fewer sentiment word groups exist in a sentence than noun word groups, then each sentiment word group must be paired with a different noun group. Finally, the group of pairs with smaller total intra-pair distance is returned as mapping between noun groups and sentiment word groups. For intra-pair distance, we consider the number of words that exist between the aspect term and the sentiment term of the pair.

In the last step, the aspects are extracted from noun groups and the corresponding sentiment scores are assigned to them. This process is shown in Algorithm 5. This phase takes the mappings created in the previous section and aspects list available as input. For each noun group - sentiment word mapping, firstly, aspects in

**Algorithm 1** Explicit aspect - sentiment mapping algorithm

```
Require: sentence: sentence to process, aspects_list: list of aspects, sentiment_dictionary: the list of sentiment word-score tuples,
    enhancers_dictionary: list of sentiment enhancers, shifters_dictionary: list of sentiment shifters
procedure Explicit Aspect - Sentiment Mapping
    noun_groups_in_clause ← Find noun groups(sentence)
    sentiment_word_groups_in_clause ← Find sentiment word groups(sentence)
    noun_group_sentiment_word_group_couples ← Map noun groups to sentiment word groups(
            noun_groups_in_clause, sentiment_word_groups_in_clause)
    aspect_sentiment_score_couples ← Extract aspect scores(
      sentence, noun_group_sentiment_word_group_couples, aspects_list)
     return aspect_sentiment_score_couples
```

**Algorithm 2** Find noun groups

```
procedure Find noun groups
Require: sentence: sentence to find noun groups
    noun_groups_in_clause ← {}
    last_found_aspect ← ""
    all_noun_groups ← create_all_possible_noun_groups(sentence)
    for each noun or noun_group n ∈ all_noun_groups do
        if n ∈ aspects_list then
            if last_found_aspect == "" OR is_clauses_connected(n, last_found_aspect) then
                add n to aspects_in_clause
                last_found_aspect ← n
    return noun_groups_in_clause
```

**Algorithm 3** Apply sentiment shifter algorithm

```
procedure Apply sentiment shifter
Require: clause: clause to process, (w, score): sentiment word-score tuple, shifter_dictionary: list of sentiment shifters
    for each wordclause_word ∈ clause do
        if clause_word ∈ shifter_dictionary then return (w, score * -1)
    return (w, score)
```

**Algorithm 4** Apply sentiment enhancer algorithm

```
procedure Apply sentiment enhancer
Require: clause: clause to process, (w, score): sentiment word-score tuple, enhancers_dictionary: list of sentiment enhancers
    index ← index_of_word(clause, w)
    possible_enhancer ← word_at_index(clause, index-1)
    if possible_enhancer ∈ enhancers_dictionary then return (w, score * 1.5)
    return (w, score)
```

**Algorithm 5** Find sentiment word and score algorithm

```
procedure Extract aspect scores
Require: clause: clause to process, sentiment_dictionary: the list of sentiment word-score tuples
    for each wordw ∈ clause do
        if (w, score) ∈ sentiment_dictionary then return (w, score)
    return NULL
```

the noun group are extracted according to comparison with the elements in the aspect list. Hence, if a noun group $ng_i \in A$, then it is retained, otherwise $ng_i$ is discarded. By this way, all possible pairings between explicit aspects and sentiment words and their scores, $(a_i, s_j, sentimentscore)$, are generated.

*3.2.2 Extracting Implicit Aspects.* Some sentiment words not only state sentiment orientation but also imply a specific aspect as well. For example, the sentiment word *expensive* can be used without an explicit aspect word, it is known that this word is used to state negative orientation on *price*. Such aspects that are implied by sentiment words are called as *implicit aspects*.

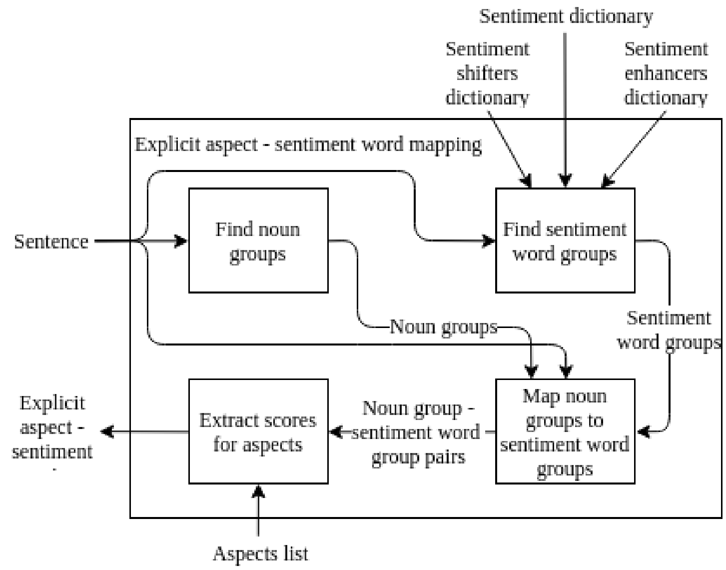


**Figure 1: Explicit aspect - sentiment mapping steps**

**Algorithm 6** Create sentiment word-aspect couples algorithm

```
procedure CREATE SENTIMENT WORD-ASPECT COUPLES
Require: clause_list: list of clauses
    sentiment_word_aspect_count_list ← {}
    for each clausec ∈ clause_list do
        sentiment_word ← find_sentiment_word(c)
        aspects_list ← find_aspects(c)
        for each aspectaspect ∈ aspects_list do
            aspect_count_list ← sentiment_word_aspect_count_list[sentiment_word]
            if aspect_count_list == NULL then
                aspect_count_list ← {}
            count ← aspect_count_lists[aspect]
            if count == NULL then
                count ← 1
            else
                count ← count + 1
            aspect_count_lists[aspect] ← count
    return sentiment_word_aspect_count_list
```

**Algorithm 7** Extract hidden aspect algorithm

```
procedure EXTRACT HIDDEN ASPECT
Require: sentiment_word_aspect_count_list: list of sentiment word-aspect count couples, sentiment_word: sentiment word with implicit
    aspect, clause_list: list of clauses
    for each clausec ∈ clause_list do
        sentiment_word ← find_sentiment_word(c)
        aspect ← find_aspect(c)
        if sentiment_word != NULL AND aspect == NULL then
            aspect_count_list ← sentiment_word_aspect_count_list[sentiment_word]
            aspect ← find_aspect_with_highest_count(aspect_count_list)
     return aspect
```

Extraction of implicit aspects is applied as an additional step after *Explicit Aspect - Sentiment Mapping*. In order to find whether a sentiment word implies an aspect or not, we rely on previous observations. This can be simply obtained from a previously collected text corpus on the domain. More specifically, in our case, we count the number of times a sentiment word is coupled with a specific aspect in the results set of explicit aspect - sentiment mapping, for each aspect. For the aspect-sentiment word mapping with the highest count, aspect is considered to be implied by the sentiment word under the following threshold values:

(1) the total number of times sentiment word is used,
(2) the ratio of the count of the aspect to total number of times sentiment word is used, and
(3) the count difference between this mapping and the second sentiment word mapping for the same aspect.

This process is applied for all sentiment words in the dataset. For all sentences containing sentiment words with implicit aspect, a mapping between the aspect and the sentiment word is added even if the aspect does not exist in the sentence. The corresponding algorithm is shown in Algorithms 6 and7.

*3.2.3 Determining Polarity of Aspect Dependent Sentiment Words.* Although most of the sentiment words have a predefined polarity value regardless of the aspects they affect, sentiment orientations of some of the words change with respect to the aspects they are coupled with. One of the most well-known examples is the sentiment word *high*. For instance, in the sentence *The price of the phone is very high*, the sentiment word *high* has a negative orientation. On the other hand, the same sentiment word is used in a positive manner in *It takes very high quality photos*. As shown in Figure 2, an additional step is applied for polarity detection of aspect dependent sentiment words. For this process, it is assumed that the list of aspect dependent sentiment words is already available, however their sentiment orientation is unknown. This phase is composed of two sub steps. In the first one, we make use of the polarity of other sentiment words in the same clause. In the second one, sentiment polarities in different clauses within the same sentence are used.

In the first sub step, neighboring sentiment words help determining the polarity score of an aspect dependent sentiment word. For example, in the sentence *The design of the phone is thick but beautiful*, *thick* is an aspect dependent sentiment word. Since the other sentiment word, *beautiful*, has positive sentiment value and negative conjunction *but* exists between *beautiful* and *thick*, the negation of the sentiment score of *beautiful* is assigned to the aspect dependent sentiment word *thick* for the aspect *design*. To fulfill this, assuming that at least one non-aspect dependent sentiment word exists in the clause, firstly, a random sentiment word other than the aspect dependent sentiment word is selected. Then, the type of the conjunction between the selected sentiment word and aspect dependent sentiment word is checked. If no negative conjunction is found between them, than the sentiment score of randomly selected sentiment word is assigned to aspect dependent sentiment word. Otherwise, this sentiment score is negated (i.e., multiplied by -1) before assigning to the aspect dependent sentiment word.

If sentiment score cannot be predicted with the help of the first sub step, sentiment words in the neighboring clause are used to predict the polarity score. For instance, the sentence *The design of the phone is not thin but screen quality is very good.* contains two sentiment word groups in neighboring clauses: *not thin* and *very good*. In addition, *thin* is an aspect dependent sentiment word. Since there exists a negative conjunction *but* between them, the negation of the sentiment score of *beautiful* is assigned to sentiment word group *not thin*. Afterwards, due to the sentiment shifter *not*, this sentiment score is again multiplied by -1 and the same sentiment score with *beautiful* is given to the aspect dependent sentiment word *thin* for the aspect *design*.

By assuming there exists at least two neighboring clauses with sentiment word groups, one of them is selected randomly and the type of the conjunction word between the neighboring clauses is checked. If a positive conjunction word is found between them, then the sentiment score of the nearest sentiment word of the selected sentiment word group is used by considering sentiment shifter, if exists. Then, this score is directly assigned to aspect dependent sentiment word, if there is no sentiment shifter affecting aspect dependent sentiment word. Otherwise, it is multiplied by -1 before assigning. On the other hand, if a negative conjunction word exists between, after considering any sentiment shifter, sentiment score is negated and then assigned to aspect dependent sentiment word. The corresponding algorithm for aspect dependent sentiment score detection is given in Algorithm 8.

## 4 EXPERIMENTS AND EVALUATION RESULTS

For the evaluation, we crawled the the product reviews on a specific mobile phone model posted at Donanim Haber [1], which is one of the most popular technology news portal in Turkey. The complete dataset contains more than 500000 sentences. However, due to difficulty for constructing ground truth over the whole dataset, we selected a random sample of 1000 sentences and manually annotated them.

Detailed statistics of this data set is presented in Table 1.

For explicit aspect sentiment matching, about 91% of aspect-sentiment matchings contain correct aspect-sentiment matching, resulting with a high precision. As the recall value, nearly 80% of all aspect-sentiment pairs in these sentences are extracted and this results with 85% of F-score.

The precision value is equal to precision ratio reported in [3] from which explicit aspect-sentiment extraction idea is adopted from. However, higher recall value is achieved in [3]. There are several reasons for why recall value is not as high as precision. Firstly, there are some aspect-based sentiment mapping cases that are out of scope of this work: co-reference resolution, comparative sentence, irony detection etc. Hence, aspect-sentiment pairs that belong to such cases cannot be detected, and this leads to decrease in recall value. In addition, the informal language used in the dataset causes difficulty for proper extraction of words, as well.

In order to extract implicit aspects, as the corpus, the complete dataset is used. However, the evaluation is conducted on the same sample of 1000 sentences chosen randomly in the previous experiment.

[1] http://donanimhaber.com

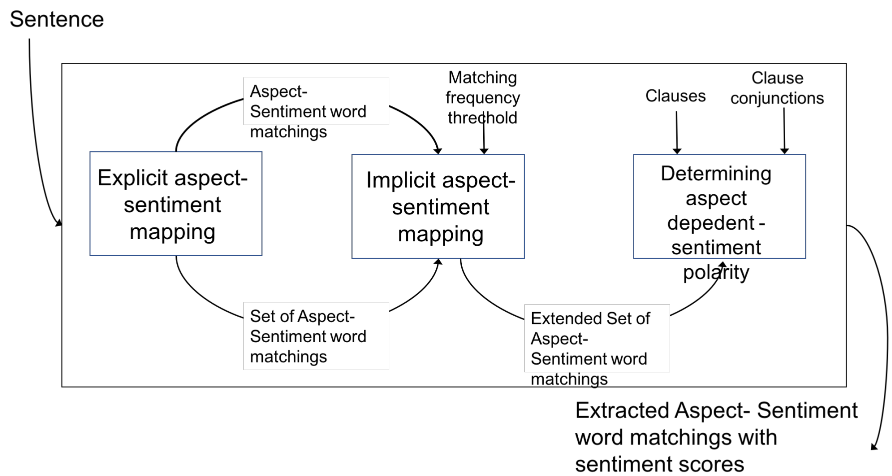


**Figure 2: Overall aspect - sentiment matching process**

---

**Algorithm 8** Aspect Dependent Sentiment Word Classification algorithm

**procedure** ASPECT DEPENDENT SENTIMENT WORD CLASSIFICATION

**Require:** *clause*: clause to classify, *neighbor_clauses*: list of neighbor clauses of main clause, *conjunction_dictionary*: the list of conjunction-score tuples, *aspect_depentdent_sent_words_dictionary*: the list of aspect dependent sentiment words

  **for** each word $w \in$ *clause* **do**
    **if** $w \in$ *aspect_depentdent_sent_words_dictionary* **then**
      *aspects_in_clause* ← Find aspects(clause, aspects_list)
      **if** *aspects_in_clause* is not empty **then**
        **for** each clause $nc \in$ *neighbor_clauses* **do**
          **if** is_connected_with_conjuction(clause, nc, conjunction_dictionary) **then**
            *score_of_nc* ← classify_phrase(nc)
            **if** *score_of_nc* $\neq NULL$ **then**
              *score_of_conjunction* ← get_score_of_conjuction(clause, nc, conjunction_dictionary)
              *score_of_aspect_dependent_sent_word* ← *score_of_nc* ∗ *score_of_conjunction*
              Save(*score_of_aspect_dependent_sent_word,* aspects_in_clause, w)
  **return** NULL

---

**Table 1: Explicit aspect-sentiment mapping results**

| Property | Count |
|---|---|
| Processed sentences | 1000 |
| Sentences without aspect-sentiment pair | 92 |
| Total number of aspect-sentiment pairs | 1169 |
| Sentences with aspect-sentiment pair | 908 |
| Total number of correctly mapped aspect-sentiment pairs | 863 |
| Total number incorrect aspect-sentiment mapping | 223 |
| Total number incorrect sentiment score extraction | 83 |

In the dataset, three frequent sentiment word - implicit aspect pairs are used: (expensive-price), (effordable-price) and (cheap-price). On the other hand, extracting implicit aspects extraction method returned three sentiment word - implicit aspect couples: (expensive-price), (efforable-price) and (original-rom). Therefore, both precision and recall ratios for the implict aspect extraction method are 66.67%.

The precision, recall and F-score values reported in [6] vary between 70-75%. However, there are inherent advantages for the study. First of all, the language of the data sets are different. Morphologically rich structure of Turkish language may cause challenges in the morphological analysis. In addition, the there is difference in the type of the text used, an informal text collection is used in the proposed study.

**Table 2: Aspect dependent sentiment word polarity prediction results**

| Mapping | Score[-4,+4] | Frequency |
| --- | --- | --- |
| camera-big | -2.5 | 52 |
| screen-big | 2.1 | 155 |
| screen-small | -2.4 | 93 |
| language-small | -2.0 | 7 |
| case-thick | -2.8 | 23 |
| phone-thick | -2.4 | 87 |
| battery-long | 2.1 | 54 |
| material quality-long | -2.0 | 10 |
| performance-long | 2.9 | 29 |
| cover-thin | 2.5 | 35 |
| box-thin | 2.0 | 26 |
| phone-thin | 2.3 | 50 |
| design-thin | 2.3 | 18 |

Under implicit aspect extraction, precision ratio decreases slightly from 91% to 90%. The reason for this decrease is that although sentiment words with implicit aspects are used to classify implicit aspect most of the time, they may be used to classify other aspects, too. Matching all occurrence of the sentiment word with implicit aspect to corresponding aspect causes several incorrect mappings and decreases precision. On the other hand, the recall value increases from 79.5% to 81%, which is more than the drop in precision ratio. These results show that the F-score obtained under implicit aspect extraction, which is 85.4%, approaches to F-score achieved in [3].

For the analysis on aspect dependent sentiment words, six different aspect dependent sentiment words are used: *big, small, short, long, thin, thick.*

During classification of aspect dependent sentiment words, as the corpus, all the dataset, approximately 500000 sentences, are used. Firstly, for each different aspect dependent sentiment word - aspect pair existing in the sentences, list of all predicted scores for this pair are extracted. In total 1123 scores are assigned to aspect dependent sentiment words - aspect pairs. After that, firstly, aspect dependent sentiment words - aspect pairs whose predicted score list contains less than 5 predicted scores is eliminated. Secondly, the pairs whose predicted score list does not contain scores in one orientation with at least 75% are also discarded. The average of remaining 639 scores for corresponding aspect dependent sentiment words - aspect pairs can be seen in Table 2.

Although, there exists two incorrect aspect dependent sentiment word - aspect pairs, which are material quality-long and language-small, the other 11 aspect dependent sentiment word - aspect pairs are useful. Moreover, it can be seen that frequencies of these two meaningless pairs are much lower than frequencies of the other pairs. In addition, the sentimental orientation of predicted sentiment scores for these 11 meaningful pairs are correct.

In order to measure precision and recall performance under aspect dependent sentiment word polarity values, the same data set of 1000 sentences are used. Table 3 presents the results after sentences which contains aspect dependent sentiment words are remapped with predicted sentiment scores for aspect dependent sentiment word - aspect pairs predicted. It can be interpreted that precision does not change when aspect dependent sentiment words are used. The reason for this situation is that the meaningless aspect dependent sentiment word - aspect pairs leads mismatching between sentiment word groups and noun groups; therefore, precision can not be increased. On the other hand, improvement is obtained for recall ratio.

## 5 CONCLUSION

A crucial step in the aspect-based sentiment analysis is aspect-sentiment mapping which connects aspects with sentiment words that affect them. In the proposed method, firstly, the sentiment word groups and noun groups are extracted from sentences and they are matched to each other. After that, for each aspect in the noun groups the sentiment score of corresponding sentiment word group is calculated and mapping between aspects and sentiment scores is created. In order to increase the mapping accuracy, implicit aspects are extracted from sentiment words referring them with the help of previously extracted aspect-sentiment word pairs. Finally, the sentiment scores of aspect dependent sentiment words are predicted by using sentiment scores of the other sentiment words. The explicit aspect - sentiment word mapping technique lead to useful level results at the basic level. These results are further improved by the proceeding two steps for refining the result.

Some improvements can be proposed for the aspect-sentiment mapping. The grammar of Turkish language can be analyzed ad exploited for improving the mapping. For example, by analyzing the positions and effects of sentiment shifter words in Turkish sentences intensively, final sentiment scores can be calculated more precisely. Furthermore, different methods on noun group - sentiment word group matching can be developed. For example, sentence parse tree can be generated from a given sentence, then the distances in this tree can be used for matching.

## 6 ACKNOWLEDGEMENTS

This work is supported by San-Tez program with project number 0740.STZ.2014.

## REFERENCES

[1] Ahmet Afsin Akın and Mehmet Dündar Akın. 2007. Zemberek, an open source NLP framework for Turkic Languages. *Structure* 10 (2007), 1–5.

[2] Sasha Blair-Goldensohn, Kerry Hannan, Ryan McDonald, Tyler Neylon, George A Reis, and Jeff Reynar. 2008. Building a sentiment summarizer for local service reviews. In *WWW Workshop on NLP in the Information Explosion Era*, Vol. 14. 339–348.

[3] Xiaowen Ding, Bing Liu, and Philip S Yu. 2008. A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*. ACM, 231–240.

[4] Magdalini Eirinaki, Shamita Pisal, and Japinder Singh. 2012. Feature-based opinion mining and ranking. *J. Comput. System Sci.* 78, 4 (2012), 1175–1184.

[5] Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*. Association for Computational Linguistics, 241–248.

[6] Zhen Hai, Kuiyu Chang, and Jung-jae Kim. 2011. Implicit feature identification via co-occurrence association rule mining. In *Computational Linguistics and Intelligent Text Processing*. Springer, 393–404.

[7] Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 168–177.

[8] Batuhan Kama, Murat Ozturk, Pinar Karagoz, Ismail Hakki Toroslu, and Ozcan Ozay. 2016. A Web Search Enhanced Feature Extraction Method for Aspect-Based Sentiment Analysis for Turkish Informal Texts. In *Big Data Analytics and*

**Table 3: Aspect-sentiment matching results**

| Case | Precision | Recall | F-score |
|---|---|---|---|
| Explicit aspect - sentiment mapping | 91.22 | 79.47 | 84.95 |
| Implicit aspect extraction | 90.36 | 80.89 | 85.36 |
| Aspect dependent sentiment word classification | 90.39 | 83.41 | 86.75 |

*Knowledge Discovery - 18th International Conference, DaWaK 2016, Porto, Portugal, September 6-8, 2016, Proceedings*. 225–238.

[9] Bing Liu. 2012. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies* 5, 1 (2012), 1–167.

[10] Qiaozhu Mei, Xu Ling, Matthew Wondra, Hang Su, and ChengXiang Zhai. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *Proceedings of the 16th international conference on World Wide Web*. ACM, 171–180.

[11] Ana-Maria Popescu and Orena Etzioni. 2007. Extracting product features and opinions from reviews. In *Natural language processing and text mining*. Springer, 9–28.

[12] G Vinodhini and RM Chandrasekaran. 2012. Sentiment analysis and opinion mining: a survey. *International Journal* 2, 6 (2012).

[13] Xiaojun Wan. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 553–561.

[14] Wei Wei and Jon Atle Gulla. 2010. Sentiment learning on product reviews via sentiment ontology tree. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, 404–413.

[15] Janyce M Wiebe, Rebecca F Bruce, and Thomas P O'Hara. 1999. Development and use of a gold-standard data set for subjectivity classifications. In *Proceedings of the 37th annual meeting of the Association for Computational Linguistics on Computational Linguistics*. Association for Computational Linguistics, 246–253.

[16] Hong Yu and Vasileios Hatzivassiloglou. 2003. Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*. Association for Computational Linguistics, 129–136.