# Feature Extracted Sentiment Analysis of Customer Product Reviews

Neena Devasia
Department of Computer Science
TKM College of Engineering
Kollam, India

Reshma Sheik
Department of Computer Science
TKM College of Engineering
Kollam, India

***Abstract*—Online shopping is more and more common nowadays. The growth in its popularity has led to increase in customer reviews that a product receives. A customer who has to choose the right product among the huge varieties of products, depends heavily on the product reviews to make a purchase decision. With great volume of product reviews, it become difficult for customers to wade through all reviews to make an informed product choice. Nowadays customers look for features that can serve them specifically. But from the thousands of reviews, it is practically impossible for customers to identify the reviews which speak about the specific product feature. As a solution to these problems, in this work we aim to analyze a product at feature level, from the customer product reviews. The proposed system, follows a semantic based approach to extract product features. An algorithm, which employ typed dependencies, is introduced for this purpose. Recursive Deep model is used to identify sentiment orientation of review sentences. A review matrix is constructed to find the importance and polarity of each product feature. The experimental results show that the method proposed is effective and has achieved the desired objective.**

***Keywords*** - *Product reviews, Feature extraction, Sentiment orientation, Dependency relations, Recursive deep model*

## I. Introduction

Popularity of online shopping has grown drastically over the years. This growth is the result of various merits, online shopping has over the conventional shopping practice. Some of the noted merits are, the time it saves to buy a list of products with few mouse clicks, freedom to determine affordable item by a straightforward comparison of prices, 24/7 availability of online stores and the comfort of the office or home. But it is difficult to get help from the professional sales staff to buy a product. To deal with this, merchant started providing meta-data for the products sold online. For customers, it was difficult to make decision about the product, with the details product speak about itself. As a solution to this, online merchants enabled forums, which allow customers to express their opinions and to get reviews.

Consumer reviews are significantly more trusted than the descriptions that come from manufacturers. This is because, a consumer review serves to explain what the product is about and how it works. Even the badly written reviews will still give the reader some information on the products features and functions. Those that are better written will even show the potential buyer the benefits of using this product over another or perhaps pointing a negative features. Consulting reviews is now a logical step in the purchasing cycle for all types of products and services.

There are two main problems that hinder customers from fully utilizing the reviews. First one is the explosive growth of reviews. It becomes hard for the customers to read all the reviews, to make a purchase decision and he may get a biased view about the product, if he reads only a few of those reviews. This may even adversely affect the customer purchases. Second problem is that, in reality, consumers like to think of what they buy as being personalized for them and look for features that can serve them specifically. But from the thousands of reviews, it is practically impossible for customers to identify the reviews which speak about the specific product feature.

In this work, we solve the above issues, by analyzing the reviews based on product features. The key module of this system is the product feature extraction module, which extracts product features from unstructured reviews. We propose a new algorithm, which extract product features using the combinations of dependencies. Stanford dependency parser is used to identify dependencies in a sentence. For finding opinion of review sentence, Stanford deep analyzer is used. A review matrix is constructed, which is used to find importance and polarity of product feature.

The rest of the paper is organized as follows. Section 2 presents some related works regarding product feature extraction and identification of sentiment polarity of product reviews. Section 3 introduces the proposed product feature extraction method and method followed to determine sentiment polarity of product features. The following section describes experimental evaluation criteria's and results. Finally, section 5, gives a conclusion of this work and discusses future scope.

## II. Related Works

The related works reveals that there have been various approaches for identifying product features from unstructured customer reviews. These approaches can be either dependency relation based or machine learning based. In dependency relation based approach, the grammatical relation between words in a sentence is mainly used to find product features. While in machine learning based approach, product features are assumed to be noun or noun phrases, so they are tagged

978-1-5090-3751-3/16/$31.00 ©2016 IEEE

and candidate product features are extracted by applying some machine learning algorithms.

The pioneer work in product feature extraction is done by Minqing Hu et al. They used NLProcessor, a linguistic parser to parse each review to split review text into sentences and to produce the part-of-speech tag for each word. Frequent itemsets were identified with Apriori algorithm. Finally, product features were derived by pruning frequent itemsets, removing unlikely features [1]. Kushal Bafna et al followed a very similar approach and follows the same underlying principle, product features will occur most frequently than other words or word phrases in a user reviews of a product. Therefore, association rule mining is used to extract the frequent occurring nouns. But association rule mining generates many features which don't actually represent features of the product. So they defined a probabilistic or characteristic power equation to remove all the feature candidates which are not real features [2]. Popescu and Etzioni et al used an unsupervised technique to extract product features from unstructured reviews. The technique named as OPINE is built on top of the KnowItAll Web information-extraction system. After extracting the noun phrases which have frequency greater than an experimentally set threshold, OPINEs Feature Assessor, evaluates each noun phrase and computes the probability associated with it. Based on the probability values product features are selected [3].

Zhijun Yan et al proposed a novel integrative approach by combining an extended PageRank algorithm that leverages relationships between product features and sentiment terms. Specifically, they pre-processes online consumer reviews using a lexical analysis tool, then constructs a network based on term pairs of candidate product features and sentiment words. Next, an extended PageRank algorithm called NodeRank is applied to rank all term pairs and derive a product feature set [4]. Khan and Baharudin et al proposed hybrid dependency patterns for features identification. The hybrid pattern is a combination of dependency patterns, which are based on dependency relation between opinion terms presented by subjective adjectives and product features presented by noun [5].

For identifying opinions of product features, most of the above discussed works considered the orientation of opinion words. Kushal Bafna et al, extracted all the adjectives (opinion words) with respective to product reviews. Their semantic polarities (positive /negative) are detected. Opinion words identified in the above process is assigned to the nearest feature in the sentence. The assignment of the opinions to the feature is achieved by computing the distance of each opinion word to detected features in a sentence and then assigning an opinion to the feature to which it is most nearest [2]. Minqing Hu et al also considers adjectives as opinion words. Adjectives share the same orientation as their synonyms and opposite orientations as their antonyms. They use this idea to predict the orientation of an adjective [1]. OPINE uses syntactic dependencies to identify opinion words. OPINE applies some extraction rules in order to find a potential opinion word. After identifying the semantic orientation of opinion words, it is assigned to the associated features to find semantic orientation of the features [3].

## III. Proposed Technique

Figure 1 gives the architectural overview of the proposed technique and it includes the following main tasks:

- Extraction of dependency relations
- Identifying candidate product features
- Pruning of candidate set
- Finding polarity of review sentences
- Finding importance of product features
- Feature sentiment analysis

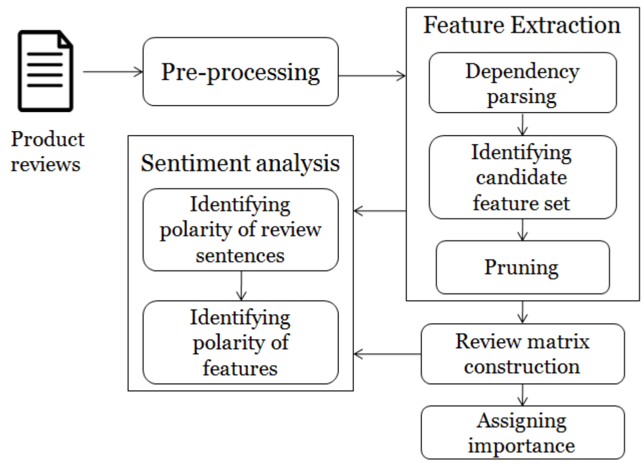


Fig. 1. Proposed System Architecture

### A. Preprocessing

The preprocessing involves identification of sentence boundary. This is because the Stanford parser expects a sentence as an input to generate typed dependencies as well as recursive deep analyzer also expect a sentence as input to find the sentiment orientation. This can be achieved using regular expressions. Other preprocessing step include removal of extraneous characters such as special symbols.

### B. Feature Extraction

The preprocessed review sentences are initially POS tagged. POS tagging is necessary for both extraction of dependency relations and identification of product features. The tagged review sentences are inputs for the dependency parser. Dependency parser extracts typed dependencies. Dependencies along with it's tags are given as input to the feature extraction algorithm, which give candidate feature set as output.

*1) Dependency relation extraction:* As noted above, for extracting dependency relations, parser requires tagged review sentence as input, hence part-of-speech tagging is crucial. POS tagger provided by stanford core nlp is used to parse each review sentence and to produce the part-of-speech tag for each word (whether the word is a noun, verb, adjective, etc).

978-1-5090-3750-6/16/$31.00 ©2016 IEEE

Dependency syntax describes sentence structures with a set of dependency relations. A dependency relation is an asymmetric binary relationship between two terms in a sentence. There can be different dependency relations in sentences, but only some of them are helpful for identifying product features. Stanford dependency parser defines approximately 50 grammatical relations [6]. Among these few of them are used in this work(see algorithm). The output from the parser for the sentence 'This is a great camera', looks like:

[nsubj(camera-5, this-1), cop(camera-5, is-2), det(camera-5, a-3), amod(camera-5, great-4)]

Where 'det', 'nsubj' and 'cop' represents the dependency relation between each of word present in the given sentence.

*2) Algorithm for feature extraction:* The candidate product features are extracted in an effective way using different combinations of dependencies. Proper analysis of review sentences reveals that the candidate product feature may be present either in the subject part or in the object part of a given sentence. In this work we have considered words associated with the candidate product features in a dependency relation may be an adjective or a verb or a noun. Apart from these combinations, product features can also be found as a verb and sometime along with $conj$ relation. These cases are also considered in our work.

**Algorithm :** Candidate product feature extraction

**Input**:
PS = $\{ps_1, ps_2,...\}$, set of parsed review sentences
D = $\{d_1, d_2, ...\}$, set of dependencies in a review sentence
**Output**:
F = $\{f_1, f_2,...\}$, set of candidate product features
**PROCESS**(PS, D):

```
begin
  for each ps_i in PS:
    for each d_i in ps_i:
      if d_i = 'conj' and d_i.arg1 ≠ noun
        for each remaining d_i in ps_i:
          if d_i = 'nsubj' and d_i.arg1 ≠ adjective
            extract f from combinations of 'nn', 'dobj',
            'amod' or 'nsubjpass'
            F.append(f)
          else if d_i = 'nsubj' and d_i.arg1 = adjective
            extract f from combinations of 'nn' or 'nsubj'
            F.append(f)
      else if d_i = 'conj' and d_i.arg1 = noun
            extract f from combinations of 'nn' or 'conj'
            F.append(f)
      else if none of the d_i = 'conj'
        if d_i = 'nsubj' and d_i.arg1 ≠ adjective
          if amod ∈ D, then
            extract f from combinations of 'nn' or 'amod'
            F.append(f)
          else
            extract f from combinations of 'nn', 'acomp'
            or 'amod'
            F.append(f)
        else if d_i = 'nsubj' and d_i.arg1 = adjective
            extract f from combinations of 'nn', 'nsubj'
            F.append(f)
return F
```

We can see in the above algorithm, for feature extraction, relation 'nn' is considered with in every case. This is to extract features in bigrams. Algorithm basically checks for the presence of noun in 'conj' and adjective in 'nsubj' relations for extracting features. The candidate feature set generated by this algorithm are not all genuine features. Thus we go for pruning.

*3) Pruning candidate feature set:* Same feature word can be in different forms. Hence, we apply stemming so that all feature words that are in plural form are converted into singular form. After stemming, features whose occurrence is greater than a threshold value is selected. Even then some common words such as 'I', 'It', 'have' etc.. can be found in the filtered set. Such words are removed to get the final product feature set.

### *C. Finding polarity of review sentences*

The objective here is to find the sentiment polarity of each review sentence i.e., whether a review sentence is positive or negative. This is achieved with Recursive Deep model. The advantage of Recursive Deep model with contrast to other sentiment prediction systems is that, it considers the order of words and builds up a representation of whole sentences based on the sentence structure, thus attaining higher accuracy than other existing systems [7, 10]. Comparison results of SentiWordNet and Deep analyzer are shown in section 4.

### *D. Finding importance of product features*

The input for this step is the review sentences and product features. Let there be total of m review sentences for a particular product, from which n features are extracted. We construct a review matrix of order of m X n using the Algorithm 2 [8].

**Algorithm 2.** Review matrix construction

**Input**:
R = $R_1$, $R_2$,... Review sentences set for a product
F = $f_1$, $f_2$,... Product feature set
**Output**:
$M_{ij}$, Review matrix
**PROCESS**(R,F):

```
begin
  for each review R_i in R:
    for each feature f_j in F:
      if f_j is present in R_i then M_ij = 1
      else M_ij = 0
return M_ij
```

A sample review matrix constructed using Algorithm 2 is given in the Table 1.Importance of a product feature is determined based on the weight assigned to it. The assignment of weights to the features depend on the frequency of occurrence of each feature in the review set. The frequency of occurrence is calculated by column sum $\Sigma M_{ij}$ for each feature $f_i$ as shown in the Table 1.

TABLE I
REVIEW MATRIX

| **Reviews** | $f_1$ | $f_2$ | $f_3$ | .. | $f_n$ |
|---|---|---|---|---|---|
| $R_1$ | 0 | 1 | 0 | - | 0 |
| $R_2$ | 0 | 0 | 1 | - | 0 |
| $R_3$ | 1 | 0 | 0 | - | 0 |
| $R_4$ | 0 | 0 | 0 | - | 1 |
| $R_5$ | 0 | 0 | 0 | - | 1 |
| $R_6$ | 0 | 0 | 1 | - | 0 |
| $R_7$ | 0 | 1 | 0 | - | 0 |
| : | - | - | 0 | - | - |
| $R_m$ | 1 | 0 | 0 | - | 0 |
| $Total$ | $\Sigma M_{i1}$ | $\Sigma M_{i2}$ | $\Sigma M_{i3}$ | - | $\Sigma M_{in}$ |

Based on frequency of occurrence of product features $\Sigma M_{ij}$ and total number of review sentences , weight $W_j$ of each feature $f_j$ is computed. Here total number of review sentence indicate number of review sentences which actually comment on product features. The equation is:

$$\frac{frequency\ \ of\ \ f_i,\ \ (\Sigma M_{ij})}{Total\ \ no.\ \ of\ \ reviews} = \left\{ \begin{array}{l} > 0.6, \quad W_j = 3 \\ 0.4 \ \text{ to } \ 0.6, \quad W_j = 2 \\ \text{Otherwise} \quad W_j = 1 \end{array} \right\} \qquad (1)$$

Determining importance of the product feature $f_j$ is based on weights $W_j$ assigned to them. The criteria followed is :

- If $W_j$ = 3, then the product feature is Very Important.
- If $W_j$ = 2, then the product feature is Moderately Important.
- If $W_j$ = 1, then the product feature is Of Little Importance.

### *E. Feature sentiment analysis*

The aim here is find whether a product feature has positive opinion or negative opinion. This can be easily analyzed from the Review matrix constructed. From Review matrix we can find which review sentence comment on a particular product feature. We have already determined polarity of each review sentence in above section. With these two information number of positive and negative reviews for each product feature is identified. We apply the following criteria to find the sentiment polarity of a product feature. Let PR and NR denotes number of positive and number of negative review sentences for a particular product feature and if P denotes it's polarity, then

$$\begin{array}{l} \text{If} \quad \dfrac{PR}{PR+NR} > 0.6, \quad \text{then} \quad P \quad \text{is +ve} \\ \text{If} \quad \dfrac{NR}{PR+NR} > 0.6, \quad \text{then} \quad P \quad \text{is -ve} \\ \qquad\qquad \text{Else} \quad P \quad \text{is neutral} \end{array} \qquad (2)$$

## IV. EXPERIMENTAL EVALUATION AND RESULTS

The proposed system has been implemented in Java. The evaluation has been done from two perspectives:

- The effectiveness of feature extraction
- The effectiveness of sentiment prediction of review sentences

In our experiment we used data set that are collected by Hu and Liu [12]. This data set contains reviews obtained from amazon.com. We tested our proposed approach using customer reviews for 3 products: Nokia 6610, Canon G3 and DVD player. Each of these product reviews consists of 700, 900 and 1050 review sentences respectively.

Accuracy is used as the measure to evaluate the performance of product feature extraction method. The definition of accuracy in this case is as follows:

- Accuracy is defined as the number of correctly mined product features(CF) divided by total number of mined product features (TF):

$$Accuracy = \frac{CF}{TF} \qquad (3)$$

The performance of our method and EXPRS, extended page rank method is compared for all 3 products and the results are shown in Table 2. The results show that our method give higher accuracy rates when compared to EXPRS.

TABLE II
COMPARISON RESULT FOR PRODUCT FEATURE EXTRACTION

| **Accuracy for** | **Nokia 6610** | **Canon G3** | **DVD Player** | **Average** |
|---|---|---|---|---|
| **Our approach** | 81.6 | 80.1 | 80.7 | 81.8 |
| **EXPRS** | 79.7 | 77.5 | 77.8 | 78.3 |

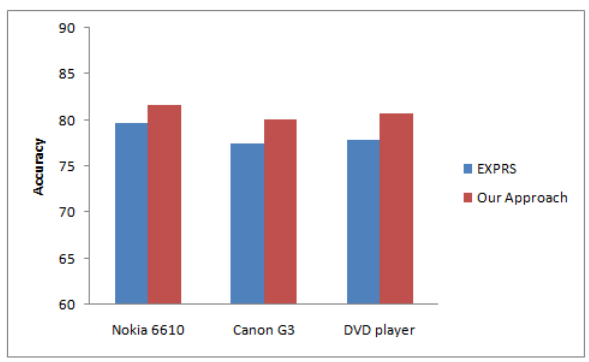


Fig. 2. Comparison of Accuracy for Proposed method and EXPRS

The above results are for explicit product feature extraction. These results can be further improved if implicit features are extracted, which is not covered in this work.

To evaluate the performance of the sentiment orientation of review sentences, we have adopted precision, recall and accuracy. A True Positive (TP)(likewise Negative (TN)) refers to an object whose actual class is positive (negative) and the sentiment analysis system predicted it to be positive (negative). False Negative (FN) classes correspond to those objects whose actual is positive and the sentiment analysis system predicted them to be negative. Similarly, the true label of False Positive (FP) class is negative and the sentiment analysis system predicted them to be positive. The definitions of precision, recall and accuracy are as following.

- Accuracy is defined as the overall success of the sentiment analysis system, i.e. is the average number of correct predictions divided by the number of all labelled review sentences:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{4}$$

- Precision is defined as the number of true positives (TP) over the number of true positives plus the number of false positives (FP):

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

- Recall is defined as the number of true positives (TP) over the number of true positives plus the number of false negatives (FN):

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

We have compared the sentiment analysis system, Recursive Deep model (RDM) with SentiWordNet. Two sets of labelled review sentences are used for analysis. One set consists of 25 review sentences and the other consist of 50 review sentences. The results are shown below.

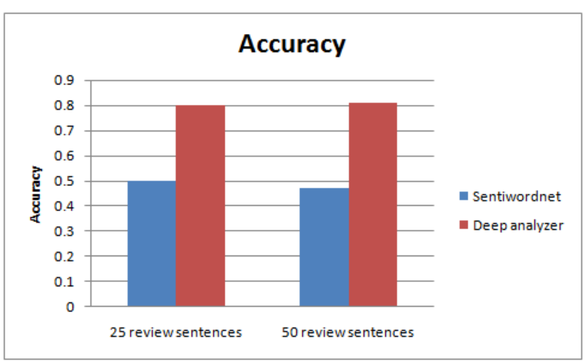


Fig. 3. Comparison of Accuracy between RDM and SentiWordNet

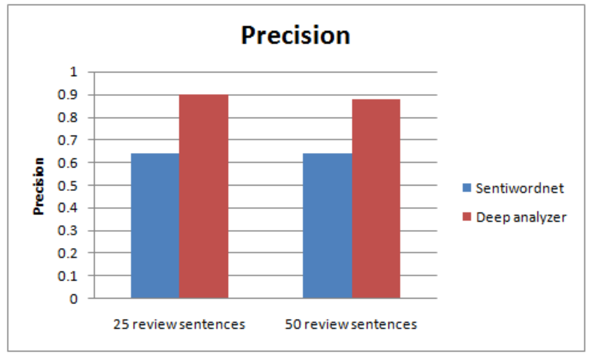


Fig. 4. Comparison of Precision between RDM and SentiWordNet

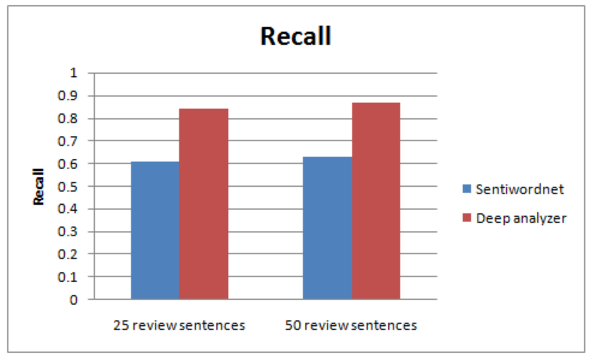


Fig. 5. Comparison of Recall between RDM and SentiWordNet

Some of the product features extracted, it's polarity and importance value for input data set of Nokia 6610 is shown in Table 3.

TABLE III
SOME OF THE FEATURES EXTRACTED FOR NOKIA

| **Features** | **Polarity** | **Importance** |
|---|---|---|
| *Batterylife* | Neutral | Moderately Important |
| *Camera* | Positive | Moderately Important |
| *Bluetooth* | Positive | Very Important |
| *Keypad* | Negative | Of Little Importance |
| *Size* | Negative | Of Little Importance |
| *Look* | Positive | Moderately Important |
| *Screen* | Positive | Moderately Important |
| *Phone* | Neutral | Moderately Important |

## V. CONCLUSION

In this work, we have presented a different method to extract product features from customer product reviews. The objective is to provide a feature based opinion of a large number of customer reviews of a product sold online. Experimental results indicate that the algorithm introduced is very promising and techniques used are efficient in performing their tasks. The problem, analysis of customer product reviews, will become increasingly important as more people are buying and expressing their opinions on the Web. Analyzing the opinions of reviews is not only useful to common shoppers, but also crucial to product manufacturers.

The techniques used can be further improved and refined. Product feature extraction can be further improved by adding implicit feature extraction. Some review sentences only include sentiment words without associated feature terms. The product features that do not appear in review sentences but are actually referred to are called implicit features. In this work, sentiment of review sentences are computed solely using Recursive Deep Analyser. This can further be improved using machine learning techniques, so that Deep Analysers score can be taken as one of the features for classication. Novel features for capturing contextual information can also be incorporated along with the other features used for classication.

## REFERENCES


[1] Hu, Minqing, and Bing Liu. ”Mining and summarizing customer reviews.” Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 2004.

[2] Bafna, Kushal, and Durga Toshniwal. ”Feature based summarization of customers reviews of online products.” Procedia Computer Science 22 (2013): 142-151.

[3] Popescu, Ana-Maria, and Orena Etzioni. ”Extracting product features and opinions from reviews.” Natural language processing and text mining. Springer London, 2007. 9-28.

[4] Yan, Zhijun, et al. ”EXPRS: An extended pagerank method for product feature extraction from online consumer reviews.” Information & Management 52.7 (2015): 850-858.

[5] Khan, Khairullah, Baharum Baharudin, and Aurangzeb Khan. ”Identifying product features from customer reviews using hybrid patterns.” Int. Arab J. Inf. Technol. 11.3 (2014): 281-286.

[6] De Marneffe, Marie-Catherine, and Christopher D. Manning. Stanford typed dependencies manual. Technical report, Stanford University, 2008.

[7] Deep Learning for Sentiment Analysis, http://nlp.stanford.edu/sentiment/

[8] Hiremath, P. S., Siddu P. Algur, and S. Shivashankar. ”Cluster Analysis of Customer Reviews Extracted from Web Pages.” Journal of Applied Computer Science & Mathematics 9 (2010).

[9] Kumar, Ravi V., and K. Raghuveer. ”Web User Opinion Analysis for Product Features Extraction and Opinion Summarization.” International Journal of Web & Semantic Technology 3.4 (2012): 69.

[10] Socher, Richard, et al. ”Recursive deep models for semantic compositionality over a sentiment treebank.” Proceedings of the conference on empirical methods in natural language processing (EMNLP). Vol. 1631. 2013.

[11] Survey on Online Reviews http://searchengineland.com/2013-study-79-of-consumers-trust online-reviews-as-much-as-personal-recommendations-164565.

[12] Riloff, Ellen, Janyce Wiebe, and Theresa Wilson. ”Learning subjective nouns using extraction pattern bootstrapping.” Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4. Association for Computational Linguistics, 2003.


978-1-5090-3750-6/16/$31.00 ©2016 IEEE