Building Comparative Product Relation Maps by Mining Consumer Opinions on the Web

Kaiquan Xu
City University of Hong Kong, kaiquan.xu@student.cityu.edu.hk

Stephen Shaoyi Liao
City University of Hong Kong, issliao@student.cityu.edu.hk

Raymond Y.K. Lau
City University of Hong Kong, raylau@cityu.edu.hk

Heng Tang
University of Macao, hengtang@umac.mo

Shanshan Wang
USTC-CityU, sswang@ustc.edu

Follow this and additional works at: http://aisel.aisnet.org/amcis2009

Building Comparative Product Relation Maps by Mining Consumer Opinions on the Web

Kaiquan Xu
Department of Information Systems,
City University of Hong Kong
kaiquan.xu@student.cityu.edu.hk

Stephen Shaoyi Liao
Department of Information Systems,
City University of Hong Kong
issliao@student.cityu.edu.hk

Raymond Y.K. Lau
Department of Information Systems,
City University of Hong Kong
raylau@cityu.edu.hk

Heng Tang
Faculty of Business Administration,
University of Macao
hengtang@umac.mo

Shanshan Wang
USTC-CityU Joint Research Center
sswang@ustc.edu

ABSTRACT

With the Web 2.0 paradigm, users play the active roles in producing Web contents at online forums, wiki, blogs, social networks, etc. Among these users contributed contents, many of them are opinions about products, services, or political issues. Accordingly, extracting the comparative relations about products or services by means of opinion mining techniques could generate significant business values. From the producers' perspective, they could better understand the relative strength or weakness of their products, and hence developing better products to meet the consumers' requirements. From the consumers' perspective, they could exercise more informed purchasing decisions by comparing the various features of certain kind of products. The main contribution of this paper is the development of a novel Support Vector Machine (SVM) based comparative relation map generation method for automatic product features analysis based on the sheer volume of consumer opinions posted on the Web. The proposed method has been empirically evaluated based on the consumer opinions crawled from the Web recently. Our initial experimental results show that the performance of the proposed method is promising, and the precision can achieve 73.15%.

Keywords

Web Mining, Business Intelligence, Information Extraction, Multi-class Classification, Machine Learning.

INTRODUCTION

In web 2.0, there exist large amount of user-generated contents, e.g., customer reviews, forum posts, and blogs. Mining users' opinions have become a hot research topic, for these data always contains much valuable information for business applications. The existing work mainly focused on two problems: one is to judge the sentiment polarity of a piece of opinion; the other is to extract and summarize the opinions on product features (Bing Liu, 2008). But less work was conducted on another more interesting and important problem, mining comparative relation. In customer reviews, users usually prefer to compare several similar products. For example, here is a segment from the customer reviews of *Nokia N95* on the *Amazon*:

I'd WOULD have given it 4 stars as it's already better than an iPhone with an FM radio, a better camera, and numerous other advantages.

These comparing opinions are more useful in practice: for individual users, they always refer to these opinions when they make choices among similar products; for producers, they always want to know their products' advantages and disadvantages when comparing with the similar products of other suppliers

But usually, these comparing opinions are hidden in large amount of opinions, and information retrieval technologies can not find them very well, so people have to spend lot of time and labor reading lots of texts to recognize and summarize these comparing opinions. If these comparing opinions can be automatically extracted and expressed succinctly as a tuple, like

better(Nokia 95, iPhone, camera) (which means *Nokia 95 is better than iPhone in the camera attribute*). And even, these tuples can be summarized and represented as comparative relation maps, an example in the following:
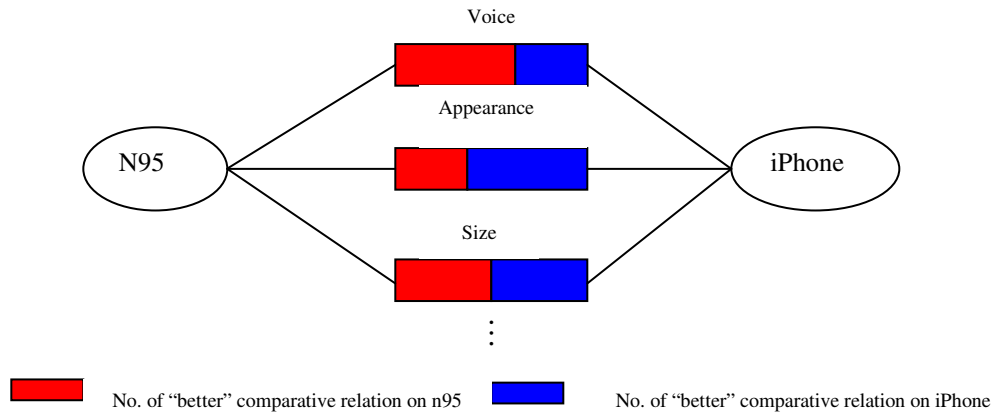

Figure 1. Product Comparative Relation Map

This comparative relation map can explicitly express, which products are better than the similar products on which product features, according to the customers' reviews. This comparative relation map will bring many benefits for product manufacturers to capture their products' strength or weakness comparing to other similar products, and for customers to choose the products of the features they are interested in.

But building the comparative relation map is quite complicated. It involves identifying product names and attribute names, recognizing and categorizing the comparative relations between product names and attribute names. Especially, the comparative relation includes more than two entities, and has several possible categories ("better", "worse" and "same"). These new characteristics make extracting the comparative relations more difficult than the traditional relation extraction in information field (Dmitry Zelenko, etc 2003).

In this paper, we first formally describe this problem, and then propose a feasible process for it. Especially, for the key and complicated step of this process: identifying and categorizing the comparative relations, we adopt the multi-class Support Vector Machine (SVM) (I. Tsochantaridis etc. 2004), and show it more effective compared with another multi-class classification method, maximum entropy model (Berger,etc. 1996). In addition, various linguistic features are evaluated on their effectiveness. The preliminary result shows the performance is quite competitive, implying the feasibility of building comparative relation map from web opinions.

This paper is organized as: Section 2 introduces the related work; the formal description of the comparative relation extraction and the proposed process are described in Section 3; Section 4 focuses on identifying and categorizing the comparative relations; Section 5 presents the initial empirical evaluation; The last section includes the conclusion and future work.

RELATED WORKS

Until now, there was little work on extracting comparative relation from web opinions. The only related research was done by Nitin Jindal etc (Nitin Jindal etc. 2006; Bing Liu etc. 2006), which mainly focused on identifying the comparative sentences and extracting relation items using Label Sequential Rule (LSR) of Part Of Speech (POS) tags. They did not differentiate the category of the comparative relation, that is, their work can not tell whether *Nokia 95* is better than *iPhone* or not. Our work classified the comparative relations into three categories: "better", "worse" and "same". In addition, although the rule-based method usually has well precision, the recall is usually low (Riloff etc. 2005). Different from previous work, the machine learning method is adopted in this paper to utilize more features to improve the performance.

Another work which is related to ours is from Shenghua etc al. (Shenghua etc. 2008). Their work mainly used some statistical measures on the key words from web pages to find competitors in some domains at a high level, such *SAP* vs *Oracle*. They did not figure out which one is better. While our work focuses on extracting comparative relations at sentence level from web opinions, and can indicate which product is better than others on certain attribute.

Another related research is relation extraction in information extraction field (Dmitry Zelenko etc. 2003), which is to recognize if there exists a special relation between two entities, such as *work_in(Tom, IBM),* which means *Tom works in IBM*. Most of work is to recognize the relation with two entities, and only to judge if the relation becomes true or not. Different from this, the comparative relation is complicated n-ary relation with more than two entities, and the relation categories have three kinds: "better", "worse" and "same". These new characteristics make it more difficult than the existing relation extraction task.

PROBLEM DESCRIPTION AND BASIC PROCESS

Extracting comparative relations from web opinions mainly involves two primary subtasks:

1) Recognizing product names and attribute names in opinions data.

For example, in this sentence,

Nokia 95 has a better camera than iPhone.

"*Nokia 95*" and "*iPhone*" should be product names, and "*camera*" should be an attribute name. For simplifying description, product names and attribute names are all treated as entities in the following sections, P_i (i=1,2,…) is used to denote the product name entity, and A_n (n=1,2,…) denotes the attribute name entity.

2) Checking if there exists a comparative relation between entities $[P_i, P_j, A_n]$, if it does, which category it belongs to. Here the comparative relation category includes "better", "worse" and "same", separately marked as ">", "<" and "=". ">" ("<") means P_i is better (worse) than P_j on the attribute A_n, and "=" means that P_i is similar with P_j on the attribute A_n

For example, in the above sentence, there is a comparative relation between [*Nokia 95*, *iPhone*, *camera*], and the relation type should be ">". But in the following sentence,

The difference between the Pearl and the Curve is the size and the keyboard.

There is not comparative relation between *[Pearl, Curve, keyboard]*.

Here a general process is presented for extracting comparative relation map from web opinions:

```
┌─────────────────┐     ┌─────────────────┐     ┌─────────────────┐
│ Collect     raw │ --> │ Annotate linguistic │ --> │ Recognize       │
│ opinion data    │     │ features        │     │ entities        │
└─────────────────┘     └─────────────────┘     └─────────────────┘
                                                          │
                                                          v
┌─────────────────┐     ┌─────────────────┐     ┌─────────────────┐
│ Summarize   and │ <-- │ Identify and classify │ <-- │ Resolve     │
│ Represent results │   │ comparative relations │   │ Pronoun     │
└─────────────────┘     └─────────────────┘     └─────────────────┘
```
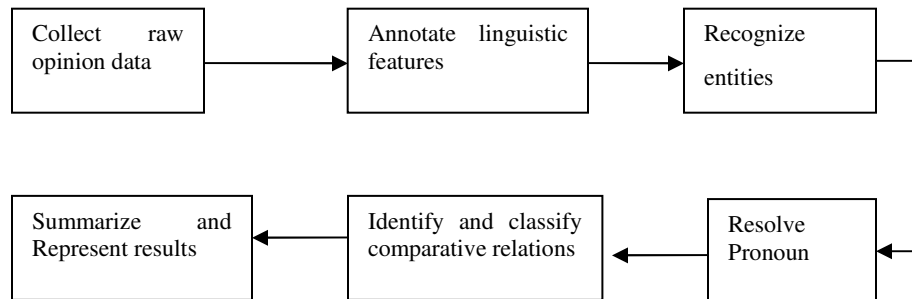
Figure 2. Process of extracting comparative relation map

1)   Collect raw opinion data: capture customer reviews from well-known online shopping sites, such as Amazon. This can be done automatically by network crawler.
2)   Annotate linguistic features: some linguistic features (such as POS tag, phrase chunking, syntactic tree, semantic role etc.) are very useful for entity recognition and comparative relation extraction, and should be annotated. This can be implemented with some Natural Language Process (NLP) tools.
3)   Recognize entities: identify and recognize product names and attribute names.
4)   Resolve Pronoun: In opinion data, there are many pronoun words (such as *it, they, them, both* etc.), which always substitute product or attribute names. They should be resolved to discover the comparative relations on them. This can be implemented by technologies from NLP field.

5) Identify and classify comparative relations: check the existence of comparative relation between entities, and recognize their categories. This is the key step, and also the most difficult one.
6) Summarize and represent results: the extracted comparative relations results can be summarized and represented in comparative relation map organized by products and attributes.

For this problem, the following challenges arise:

1). Identifying and classifying comparative relation: Compared to the traditional relation extraction problem, the big difference of this problem is: the comparative relation involves more than two entities, and the relation category contains more types, not just two ("true" or "false"). This is a typical complicated n-ary relation recognition problem with multiple possible categories. These new characteristics make this problem more complicated. So how to accurately identify and classify the comparative relation is the key issue for building comparative relation map from web opinions.

2). Abbreviation of entities: in web opinion, some product names always occur in informal abbreviation, for example, "*BlackBerry 8320*" is written as "*BB 8320*", "*Nokia N95*" as "*n95*". How to identify entity with high accuracy is very hinge for the next steps.

3) Absence of product names in some contexts: In the customer reviews on a special product, this product name is always absent. For example, in the review on "*Nokia N95*", "*The camera doesn't seem much better than that in the Curve.*", here *"Nokia N95"* is omitted.

4) Informal languages: In web opinion, the customer reviews always contain some typing errors, or informal expressions. So the robustness of your approach is also very important.

In the following section, we will focus on the first challenge.

IDENTIFY AND CATEGORIZE THE COMPARATIVE RELATION

As indicated in the previous sections, identifying and classifying comparative relation is the hinge step and also more complicated task than the traditional relation extraction. The relation in information extraction field always involves only two entities, and its objective is to check if the relation exists or not, which is a typical binary-class classification problem. Here, the comparative relation involves three entities, and the object is not only to check if the comparative relation exists, but also recognize which category it is, ">", "<", "=", or "no_comparative". (here, we treat "*no_comparative*" as a special category to indicate that there does not exist the comparative relation between entities, instead of using an additional step to check the existence.). This problem is obviously a multi-class classification problem.

In order to recognize the complicated comparative relations, two typical multi-class classification methods, multi-class Support Vector Machine (SVM) (I. Tsochantaridis. etc. 2004) and maximum entropy model (Berger etc. 1996), are considered and compared to check their effectiveness. Also the various appropriate linguistic features are explored.

Multi-class Support Vector Machine

Originally, Support Vector Machine (SVM) is only for binary-class problem. The idea of the multi-class SVM adopting winner-takes-all principle for multiple-class problem is: it trains a classifier for every category, and the predication category is assigned by the classifier with the highest output (I. Tsochantaridis. etc. 2004). This can be re-represented as:

First, define a discriminant function (also called as score function) $F(r, c, \omega) = <\omega, \Psi(r, c)>$, here $\Psi(r, c)$ is the feature represent vector of the relation r and the category c. For example, when the sentence containing the relation r includes the words "*compared with*", and c is ">", a specific element in the feature represent vector is set to 1. The parameter ω is acquired by the learning principle: for the training examples $\{(r_i, c_i)\}_{i=1}^{n}$, maximize the minimal margins between $F(r_i, c_i, \omega)$ (the score when the relation r_i has the right category c_i.) and $F(r_i, c, \omega)$ (the score when the relation r_i has other category c.). The optimization principle of this method can be formally expressed as:

$$\max_{i \in \{1 \dots n\}, c \in C \setminus c_i} \min \{\omega \cdot \Psi(r_i, c_i) - \omega \cdot \Psi(r_i, c)\}$$

Subject to: $\|\omega\| = 1$

$$\omega \cdot \Psi(r_i, c_i) > \omega \cdot \Psi(r_i, c), \text{ here } i = 1...n,\ c \in C \setminus c_i$$

Here C is the collection of all categories.

Maximum Entropy Model

The motivating idea behind maximum entropy model (Berger, etc. 1996) is that one should prefer the most uniform models that also satisfy any given constraints. For example, in the comparative relation classification problem, assuming there is a prior experience: if the word *"better"* occurs, the relation has 70% chance of being a ">" relation. So for a comparative relation with *"better"* word, maximum entropy model considers it to be ">" relation with 70% probability, and to be the other three categories with 10% probability separately.

Maximum entropy model always has exponential form:

$$P(c\,|\,r) = \frac{1}{Z(r)} \exp(\sum_i \lambda_i f_i(r,c)) \tag{I}$$

$f_i(r,c)$ is a feature function indicating the combined feature of the relation r and class c, for example, when *"better"* occurs in the relation r, and the class c is ">", the value of $f_i(r,c)$ will be 1, otherwise be 0. λ_i is a parameter to be estimated, and indicates the weight of the i-th feature. $Z(r)$ is the normalizing factor. The learning principle of maximum entropy model is to maximize the entropy of the distribution of the formula (I) on the training data. It can be proved that the maximum entropy equals to the maximum likelihood of that formula, so the global maximum can be achieved.

It is obvious that these two kinds of multi-class classification methods have very distinguished ideas and learning principles, we compare their performances for the comparative relation extraction to choose appropriate one.

Linguistic Features

In order to acquire high precision for comparative relation extraction, the appropriate linguistic features should be captured. By referring to the characteristics of the comparative relation, we design some simple and effective linguistic features as follows:

1. Entity types at different positions: since the same comparative relation can be expressed in various ways with different entity types at different positions. For example:
    1) *Nokia N95 has a better camera than iPhone.*
    2) *The camera of Nokia N95 is better than that of iPhone.*

Although these two sentences express the same comparative relation *>[Nokia N95, iPhone, Camera]*, the entity types occur with different patterns. The pattern of 1) is *Product- Attribute-Product*, but the one of 2) is *Attribute-Product- Product.* So this feature concerns about the pattern that the entity types occur.

2. Words:
    1) Words of entities: for example, *"Nokia N95"*, *"camera"*.

    2) Key words between entities: some words are good indicators for comparative relations. Here we mainly include the following words: *"in contrast to", "unlike", "compare with", "compare to", "beat", "win", "exceed", "outperform", "prefer", "than", "as", "same", "similar", "superior to", "improvement over", "better", "worse", "best", "worst", "more", "most", "less", "least"*, etc.

3. POS tags: Some kinds of Part Of Speech (POS) tag of words are also good indicators for comparative relation. For example, JJR (comparative adjective), JJS (superlative adjective), and Verb etc.

INITIAL EMPIRICAL EVALUATION

In order to evaluate the performances of extracting comparative relation, check which multi-class classification method is more appropriate, and verify the effectiveness of various linguistic features, we designed an initial empirical evaluation. In this evaluation, we mainly considered 1) what the performances of extracting comparative relation are, which method is better, and how the size of training sample influences the performances; 2) which kind of linguistic features are more effective.

Evaluation Design

Data

The original web opinion data was collected from the *Amazon* online shopping site. It is the customer reviews about several mobile phones: *BlackBerry Curve 8320, Motorola RAZR2 V8, BlackBerry Bold 9000*, and *Nokia E71*. Three students who are familiar with mobile phone were employed to manually filter out the segments containing the comparative relations, and to annotate the entities and the categories of the comparative relations. The final annotating results must reach agreement by at least two persons. In the initial research phrase of this study, a small size dataset was annotated, consisting of 79 pieces of segments, and 217 comparative relations. Among these comparative relations, 65 are ">" relations, 48 are "<" relations, 5 are "=" relations, and 99 are "no_comparative" relations.

Data Preprocess

The corpus was preprocessed with GATE (Gate 2009), a natural language process tool, to automatically capture some linguistic features. The pre-process includes splitting sentences, stemming words and labeling POS tags.

Evaluation Setting and Criteria

We evaluated and compared the performances of two kinds of multi-class classifiers under the different ratios of the training sample size to the testing sample size. By using different linguistic features, we also evaluated their effectiveness on extracting comparative relations. The precision, the ratio of the number of right categorized samples to the number of total samples, was used as the metric of this evaluation. MaxEnt (Maximum.. etc. 2009) and SVM-multiclass (Multi-class etc. 2009) software tools were revised and used in this experiment.

Preliminary Results and Discussion

Precisions under different ratios of training sample size to testing sample size

Table 1 shows the precisions of two kinds of multi-class classifiers for comparative relation extraction with different ratios of the training sample size to the testing sample size (the ratios are respectively 20:80 (20% as training samples, 80% as testing samples), 40:60 (40% as training samples, 60% as testing samples) and 50:50 (50% as training samples, 50% as testing samples)). Figure 3 is the comparing result of two methods.

Ratio	Maximum Entropy	Multi-class SVM
20:80	60.12%	64.74%
40:60	65.39%	67.69%
50:50	72.22%	73.15%

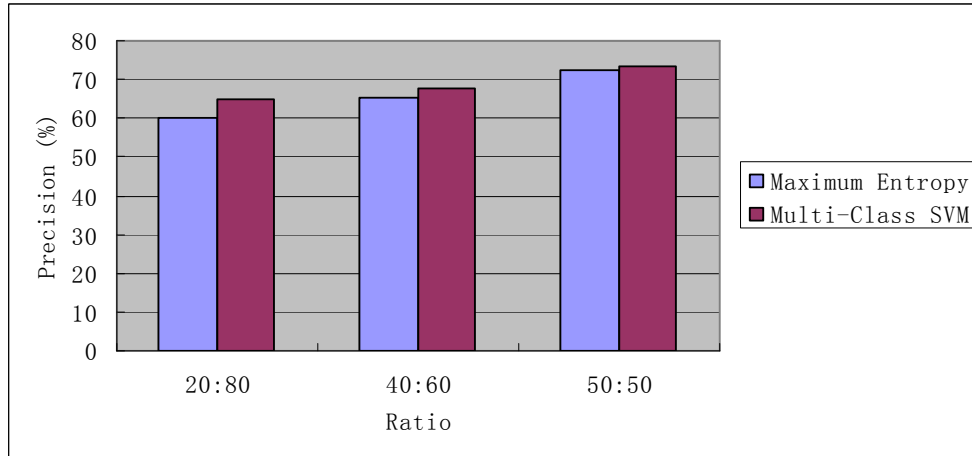**Table 1. Precisions of two classifiers with different ratios.**

Figure 3. Comparison of the performances of two classifiers.

Table 1 shows that the precision of extracting comparative relations can be more than 60%, when only 20% data is as the training samples, and it can be even more than 73%, if half of data is as training samples. The preliminary result indicates multi-class classifying methods can achieve quite competitive result in recognizing comparative relation, which implies the feasibility of building comparative relation map from web opinions.

The figure 3 shows that multi-class SVM method is better than maximum entropy model for this problem. Especially when the ratio of training sample size to the testing sample size is small, the difference is much more. But with the increasing size of training samples, the difference becomes less. This result is related with the learning principles of these two classifiers: Maximum entropy model tries to learn a classifier which satisfies the constrains from training samples and has the maximum entropy. When the training sample size is small, the constrains from training samples can not represent the real distribution of whole samples, so the performance is a little worse. But when the training samples become large, its performance improves quickly. Different from maximum entropy model, multi-class SVM tries to find a hyperplane with large margin which can separate the right samples from the wrong samples, so under small training sample size, it still has a outstanding performance.

Effectiveness of different linguistic features

In order to evaluate the effectiveness of different linguistic features, we compared the performances, when entity type, key word and POS tag features are respectively removed from the feature sets, in order to check how much the performance decline without using that type of feature. The empirical results are provided in table 2 and figure 4. (This evaluation was conducted with 50:50 as the ratio of the training sample size to the testing sample size.)

	All kinds of features	No entity type	No key words	No POS tag
Maximum Entropy Model	72.22	64.82	66.67	68.52
Multi-class SVM	73.15	65.74	67.59	68.52

Table 2. Precisions when separately removing every kind of feature.

Figure 4. Precision comparison when separately removing every kind of feature.

It can be found that for both maximum entropy model and multi-class SVM, entity type features affect the precisions mostly, key words features moderately decrease it, and POS tag features affect it least. This indicates the entity type features are most important for accurately categorizing comparative relations, and also key word features are very useful. So in order to achieve good performance in recognizing comparative relations, the entity types must be accurate, and also more key words should be explored.

CONCLUSION AND FUTURE WORK

In this paper, we describe the problem of building comparative product relation map by mining web opinions from information extraction perspective, and propose a feasible solution process for it. By exploring and comparing, the multi-class SVM is more effective for identifying and categorizing the complicated comparative relations. And the effectiveness of various linguistic features are also evaluated. The preliminary results show that the performance is quite promising, and can reach 73.15%.

Based on this initial work, in the future, first, a large scale empirical evaluation will be done using the large dataset; second, more linguistic features will be included to improve the precision of extracting comparative relations; Third, accurately recognizing product and attribute names should be studied to make sure the performance of entity recognition will not influence the performance of extracting comparative relations very much.

REFERENCES

1. Berger, Adam L. and Y, Stephen A. Della Pietra and Y, Vincent J. Della Pietra. (1996) A Maximum Entropy approach to Natural Language Processing, *Computational Linguistics*, 22, 39-71.

2. Nitin Jindal and Bing Liu. (2006) Mining Comparative Sentences and Relation, in Yolanda Gil and Raymond J. Mooney, (Eds.) *Proceedings of 21st National Conference on Artificial Intellgience*, July 16-20, Boston, Massachusetts, USA, 244 - 251

3. Nitin Jindal and Bing Liu. (2006) Identifying Comparative Sentences in Text Documents, in Efthimis N. Efthimiadis (Ed.) *Proceedings of the 29th Annual International ACM SIGIR Conference on Research & Development on Information Retrieval*, Seattle, Washington, USA   August 06 - 11, 244-251.

4. Shenghua Bao, Rui Li, Yong Yu, Yunbo Cao.(2008) Competitor Mining with the Web. *IEEE Transactions on Knowledge and Data Engineering*. 20, 10, 1297-1310.

5. I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun.(2004) Support vector machine learning for interdependent and structured output spaces. in Carla Brodley (Eds), *Proceedings of the Twenty-First International Conference on Machine Learning*, Banff, Alberta, Canada   July 04 - 08. 104.

6. GATE,(2009) A General Architecture for Text Engineering, *http://gate.ac.uk/*

7. Bing Liu.(2008) Opinion Mining and Summarization. *Tutorial at 17th International World Wide Web Conference*, Beijing, China, April 21-25

8. Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. (2003) Kernel methods for relation extraction. *Journal of Machine Learning Research*, 3, 1083-1106

9. Maximum Entropy Modeling Toolkit for Python and C++, (2009) *http://homepages.inf.ed.ac.uk/s0450736/maxent_toolkit.html*

10. Multi-Class Support Vector Machine, (2009) *http://svmlight.joachims.org/svm_multiclass.html*

11. Riloff, Ellen, Wiebe, Janyce, Phillips, William.(2005) Exploiting Subjectivity Classification to Improve Information Extraction. *Proceedings of the 20th National Conference on Artificial Intelligence*, Pittsburgh, Pennsylvania, US, July 9–13, 2005.