Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 4, Issue. 7, July 2015, pg.240 – 246

RESEARCH ARTICLE

Opinion Feature Extraction via Domain Relevance using Zipf Law

Baste Vaishnavi S. [1], Patil Dipak V. [2]

[1,2]Department of Computer Engineering, MET BKC Nashik-03

[1,2]Savitribai Phule University of Pune, Maharashtra, India

[1] vaishnavibaste07@gmail.com, [2] dipakvpatil17 @gmail.com

Abstract - With the advent of internet and technology in social networking and shopping websites, current research area is focused in field of sentiment analysis. Ample of opinion and sentiments are available in electronic text format which allow us the discovery of opinion features from online review. Existing work in the field of opinion mining extract the opinion features only from a single review dataset, disregarding the word distributional characteristics across different review dataset. To overcome this drawback, a method is proposed which identifies opinion features from two review dataset, among which one is the domain specific review dataset, and other domain independent review dataset. To make the difference between two dataset set and its opinion features, Domain Relevance is calculate, which signifies the probability of opinion feature in a given specific domain. Domain Relevance score calculated on domain specific review dataset is called Intrinsic Domain Relevance and on other domain independent review dataset is called Extrinsic Domain Relevance. Zipf law is used as a preprocessing tool which helps to eliminate stop words and give us more precise results. This cross domain classifier helps to improve opinion feature extraction as compared to existing methods.

Keywords - social networking, sentiment analysis, opinion, opinion feature, Domain relevance, cross domain.

I. INTRODUCTION

Opinions are required for all individuals when they need to make some decision while purchase of any product. Opinions expressed by human being depend a lot on human actions and choices made by others too. Our choices towards any object also depend on how others feel about the same object. Because of this reason, when we need to make a verdict we seek out the judgment and opinion of others. Organizations follow the same process and hence organizations conduct different types of information gathering through survey, interviews to know feedback from their customers [1]. Sentiment analysis with help of natural language processing works to sketch the mood of the customers about a particular product. Sentiment analysis also called opinion mining collects data from online review, inspect the data and after some processing draw conclusions on review data specifying whether negative or positive opinions are expressed in review. A sample review taken from website is given as below.

"The iPhone 6 delivers a spacious, crisp 4.7-inch screen, improved wireless speeds, better camera autofocus, and bumped-up storage capacities to 128GB at the top end. The iPhone 6 is an exceptional phone in nearly every way except its average battery life: it's thin, fast, and features the excellent iOS operating system."

The above sample review adopted from [10] expresses contradictory opinions related with different attributes or aspects of iPhone6. Camera quality, storage capacity and speed reflect the positive aspects of this phone, while battery life is annoying which gives a negative statement towards this phone. Nowadays, smart patrons are not ready to make any choice just by looking at the overall rating of the product. They want to understand why it receives the rating, that is, which are the positive aspects of the phone and which the negative aspects of the phone which contribute to the final rating of the product. It is important to exploit the exact opinion features from reviews and classify them to fine grained opinions [2].

In sentiment analysis, candidate opinion feature is an entity towards which user express their specific opinions. This method proposes an approach to the identification of such features from unstructured textual reviews.

A. Levels of Sentiment analysis

In general, sentiment analysis has been classified into below levels according to the level of granularity:

- **Document level:** The task of this level is to classify the whole document and comment sentiment on the basis of the overall document. For example, given a movie review, this level of classification just gives the overall rating of the movie. It does not classify what particular are the positive or negative aspects of the movie [3].
- **Sentence Level:** This level increases the granularity of opinion mining. The task of classification at this level goes down from document to sentences. It thus determines the opinion expressed in each sentence. Thus, it decides whether opinion expressed by sentence is positive, negative, or neutral [3].
- **Aspect level:** The above two levels of classification fails to associate what exactly customer liked or disliked about a product. This is a more fine grained classification in which instead of looking at language constructs, aspect level straightforwardly looks at the architecture [3].

II. LITERATURE REVIEW

W. Jin and H. Hay [4] proposed a Lexicalized HMM-based approach to extract opinions from online reviews. They proposed a method that extracts product entities from the product reviews available on websites. They further learn the features of the product entity and then classify the product according to the learning experience. To make system learn about the product entity, they extract opinion related to the product entity, from the product reviews they extract an opinion sentence that describes the candidate product entity and finally define extracted opinions.

N. Jakob and I. Gurevych [5] employ the supervised algorithm which represents the state-of-the-art on the employed data. They had used Conditional Random Fields (CRF) for opinion targets extraction which tackles the problem of domain portability. Their proposed work evaluates the performance of the system in cross domain.

S. M. Kim and E. Hovy [6] proposed a method extraction of opinion in the form of triplets as opinion, holder, and topic for identifying an opinion from online news media texts. In the triplet, opinion consists of the words that describe the opinion about the product. Then semantic roles are assigned to such opinion bearing word and then find the opinion holder of the product and the topic of opinion from the assigned semantic roles.

B. Pang, L. Lee and S. Vaithyanathan [7] employed document level opinion mining. Rather classifying the document on topic basis, they classified the overall sentiment determining whether a review is positive or negative. They had used this technique to classify the movie review and rate it using thumbs up and thumbs down approach. Ratings were automatically extracted and converted into one of three categories: positive, negative, or neutral.

All the above methods classify opinions within given domain and does not calculate its relevance score. Proposed method enables us to calculate domain relevance score against different domains by calculating its intrinsic and extrinsic domain relevance, which helps to give more accurate opinions results in regard with the particular domain.

III. ALGORITHM AND IMPLEMENTATION STRATEGY

We have implemented the method proposed by Zhen Hai, Kuiyu Chang, Jung-Jae Kim and Christopher Yang [1] along with Zipf law. Proposed method identifies opinion features from online reviews by estimating the variation in opinion feature statistics across two dataset, one domain-specific review set and one domain-independent review set. The proposed method captures this difference using domain relevance (DR). Domain Relevance matches the relevance of a term in a particular domain.

A list of opinion features is generated, initially candidate features are considered. Candidate features are extracted from online review set using a set of dependency rules (specified in Table I). Candidate features are mainly noun phrases in the statement. For extracted candidate features we calculate two relevant scores

based on its domain relevance (DR) score. One is defined as Intrinsic Domain Relevance score (IDR) and other is called as Extrinsic Relevance score (EDR). IDR scores and EDR scores are computed using same algorithm. When the same algorithm is applied on domain dependent review dataset it produces IDR score of the candidate features, when algorithm is applied on domain independent review dataset, it produces EDR score of candidate feature.

IDR scores give a high relativity of a particular candidate feature in that particular document. While EDR scores are frequently occurring in other review dataset. Statistical association of such candidate features having high EDR scores is good in domain independent review dataset. Finally candidate features having high IDR scores and low EDR scores are confirmed as final opinion features. Finalization of candidate features is done using thresholding approach in which candidate features having low IDR score than a specified intrinsic threshold and having high EDR score than a specified extrinsic threshold are pruned. This filtration technique gives better results.

IV. SYSTEM ARCHITECTURE

A) System Flow

1. Different English grammar rules are used to create a list of candidate features from the selected domain review dataset.
2. Apply Zipf law for pre-processing which will be used for stop word removal. This help to give more precise results.
3. For every identified candidate feature, its domain relevance score is calculated which is used to calculate IDR and EDR. IDR is Intrinsic Domain Relevance which is calculated on domain specific review set. EDR is Extrinsic Domain Relevance and is calculated on domain independent review dataset.
4. In the final step, we confirm the candidate feature which is having high IDR score and low EDR score, rest candidate features are pruned [1].

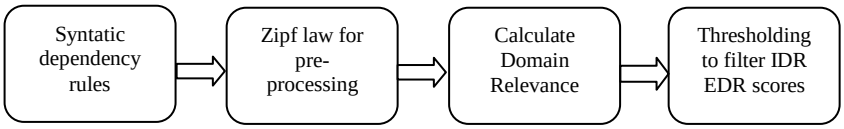

Fig 1: Proposed method workflow

B) Candidate Feature Extraction

Candidate features are extracted in the following manner: for each word, first determine if it is a noun then sequentially apply the Verb Object (VOB), Subject Verb (SBV), and Preposition Object (POB) rules. A noun which matches any of the rules is extracted as a candidate feature. Dependence Grammar discover different dependent relationship between words, different words are then combined according to dependency structure of sentences. SBV, VOB, and POB are three rules that correspond to below mentioned patterns [8]. For each relation, a rule is defined with additional restrictions for candidate feature extraction, as shown in Table 1.

TABLE I

SYNTATIC DEPENDENCY RULES

Srno.	Rule	Relationship
1	NN , SBV -> CF	Noun has a SBV relationship
2	NN , VOB -> CF	Noun has a VOB relationship
3	NN , POB -> CF	Noun has a POB relationship

The candidate feature extraction works in the following steps:

1. Dependency grammar is employed on sentence level to identify syntactic structure of the sentence.
2. The three rules in Table 1 are applied to recognize dependence structures, and the equivalent nouns are obtained as candidate features [8].

C) Methodology

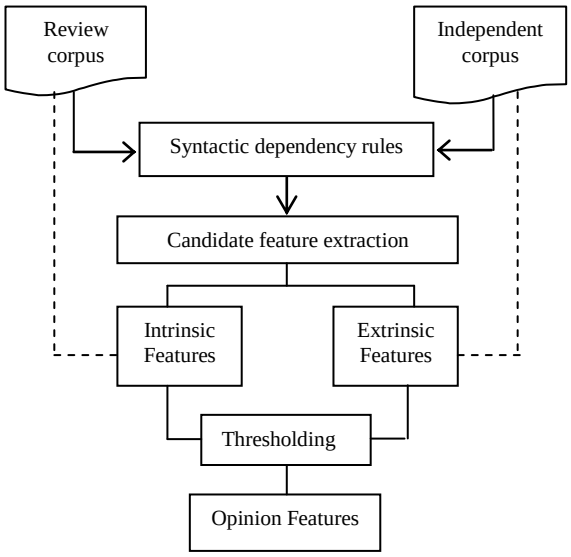

Fig 2: System Architecture

Fig2 shows the architecture of proposed method using a domain-dependent review corpus and a domain independent corpus. In first step extract a list of candidate features from the review corpus via manually defined syntactic rules. These rules must be applied to domain dependent and domain independent corpus to extract the noun features. For each extracted candidate feature, estimate its intrinsic domain relevance, which represents the statistical association of the candidate to the given domain corpus, and extrinsic- domain relevance, which reflects the statistical relevance of the candidate to the domain independent corpus. Only candidates with IDR scores exceeding a predefined intrinsic relevance threshold specified by user and EDR scores less than another extrinsic relevance threshold are confirmed as valid opinion features. In short, identify opinion features that are domain-specific and at the same time not overly generic via the inter-corpus statistics IEDR criterion [1].

V. MATHEMATICAL MODELLING

Zipf law - Zipf law governs the frequency distribution of words in a language or in a collection that is large enough to represent a language set [11]. To exemplify this law, we assume that we have

C - collection of candidate terms
V - unique words in the collection
r - rank of term
Prob(r) - probability of a term at rank r.
N - total number of terms in the collection.

For each term in the collection, we calculate

$$freq(term)$$

This calculates how many times a term occurs in a collection. Then we rank the terms in descending order by their frequency. Further, we calculate the probability of a term as -

$$Prob(r) = freq(r) / N$$

Then Zipf's law states that -

$$r * Prob(r) = A,$$

where A gives us a constant which we used for the removal of stop words from the dataset.

Domain relevance signifies how much a term is linked to a specified corpus based on two types of statistical measure, dispersion and deviation. Dispersion measures how significantly a term is cited across all documents. This is also known as horizontal significance because it measures the significance of a term all around different documents in entire dataset. Deviation intends how often a term is mentioned in a specific document. This is known as vertical significance because it is measured by distributional significance of the term in the same document.

Both dispersion and deviation are calculated using the well-known term frequency- inverse document frequency (TF-IDF) term weights.

TFij = term frequency for term Ti,
Dj = document
DFi = global document frequency

Now, weight wij of Ti in Dj is calculated as follows*:*

$$w_{ij} = \begin{cases} (1 + log\, TF_{ij}) \times log(\frac{N}{DF_i}) & if\ TF_{ij} > 0, \\ 0 & otherwise \end{cases} \quad (1)$$

where-
i = 1, 2.....M for total number of M terms
j = 1, 2.....N for total number of N documents

The standard variance s_i for term Ti is calculated as follows:

$$s_i = \sqrt{\frac{\sum_{j=1}^{N}(w_{ij} - \overline{w_i})2}{N}} \quad (2)$$

where the average weight $\overline{w_i}$ of term Ti across all documents is calculated by:

$$\overline{w_i} = \frac{1}{N}\sum_{j=1}^{N} w_{ij}$$

The dispersion $disp_i$ of each term Ti in the corpus is defined as follows:

$$disp_i = \frac{\overline{w_i}}{s_i}$$

Dispersion thus measures the normalized average weight of term T_i. It is high for terms that appear frequently across a large number of documents in the entire corpus.

The deviation $devi_{ij}$ of term Ti in document D_j is given by:

$$devi_{ij} = w_{ij} - \overline{w_j} \quad (3)$$

where the average weight $\overline{w_j}$ in the document D is calculated over all M terms as follows:

$$\overline{w_j} = \frac{1}{M}\sum_{i=1}^{M} w_{ij}$$

Deviation $devi_{ij}$ indicates the degree in which the weight w_{ij} of the term T_i deviates from the average $\overline{w_j}$ in the document D_j. The deviation thus characterizes how significantly a term is mentioned in each particular document in the corpus.

The domain relevance dr_i for term T_i in the corpus is finally defined as follows:

$$dr_i = disp_i \times \sum_{j=1}^{N} devi_{ij}$$

A. Algorithms used

Algorithm 1: Calculating Domain Relevance (IDR/EDR)

Input: A domain specific or domain independent dataset pre processed using Zipf law

Output: Domain relevance scores

for *each candidate feature term* CF_i **do**

for *each document* D_j *in the datset* C **do**

Calculate *weight* w_{ij}

Calculate *standard deviation* s_i

Calculate *dispersion* $disp_i$

for *each document* D_j *in the corpus* C **do**

Calculate *deviation* $devi_{ij}$

Compute *domain relevance* d_i

return A list of domain relevance scores for all candidate features;

Algorithm 2: Identifying opinion features via IEDR

Input: Domain Review corpus R and Domain independent corpus D

Output: A validated list of opinion features

Extract candidate feature from review corpus R;

for *each candidate feature* CF_i **do**

Compute IDR score idr_i via algorithm 1 on domain review corpus R

Compute EDR score edr_i via algorithm 2 on domain independent corpus D

if(idr_i >= ith) AND(edr_i <=eth) **then**

Confirm candidate CF_i as feature;

return A validated set of opinion feature;

VI. EXPERIMENTAL RESULTS

Experimental results are conducted on real world dataset which includes unstructured review text of various different domains: hotel, hospital, mobile phone reviews. Evaluation of IEDR performance against the competition using precision versus recall curves is shown in fig 3.

IDR - uses only given review corpus to extract opinion features.
EDR - uses only domain independent corpus to extract opinion features.
IEDR - Intrinsic and Extrinsic Domain Relevance criterion.

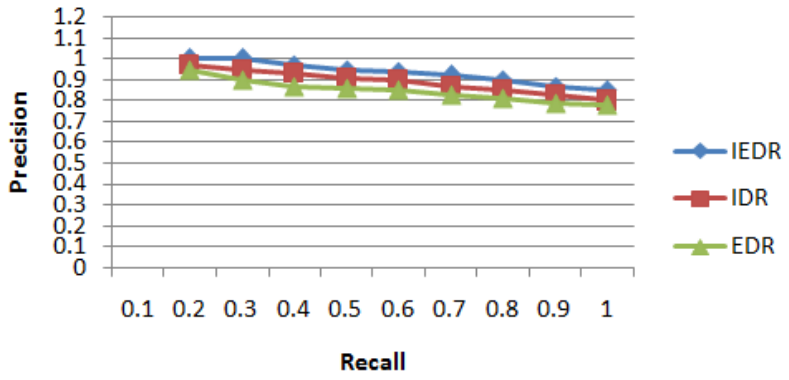

fig 3: Precision and Recall curves for hotel feature extraction

We first extracted candidate features from the given review domain, using syntactic rules defined in Table 1. Then we apply Zipf law for the removal of stop words. Based on these set of candidate feature, we compared IEDR to both IDR and EDR on hotel review domain. The precision-recall curves for IEDR, IDR and EDR are plotted as solid blue, red and green lines respectively. In fig 3 IEDR curve lies well above the IDR curve for all the recall levels. We can conclude that from the suggested methods of IDR, EDR and IEDR; IEDR gives best results with maximum precision and recall value of 1.

VII. CONCLUSIONS

Thus, with zipf law as a method for pre processing we have implemented a method for intercorpus opinion feature extraction based on the IEDR feature-filtering criterion. Zipf law gives us frequency distribution of a terms and helps us to determine the probability of a term in a document. This approach applies the distributional differential characteristics of features across two review set. It identifies candidate features that are relative to the given review domain and yet not very generic. Also, the difference between two domains is important. The more two domains are different, IEDR yield more better results.

In future work, fine grained topic modeling will employed which will improve the accuracy and efficiency of the opinion feature extraction including non-noun features, infrequent features will be considered. Current system classify only positive and negative opinions, in addition, neutral opinions will be considered for future work.

REFERENCES

[1] Zhen Hai, Kuiyu Chang, Jung-Jae Kim, and Christopher C. Yang, "Identifying Features in Opinion Mining via Intrinsic and Extrinsic Domain Relevance," *IEEE Transactions on Knowledge and Data Engineering*, Vol.26 No.3, March 2014, pp 623-634.

[2] G. Vinodhini, RM. Chandrasekaran, "Sentiment Analysis and Opinion Mining : A Survey," *Proceedings of International Journal of Advanced Research in Computer and Software Engineering*, Vol. 2, June 2012.

[3] Bing Liu, "Sentiment Analysis and Opinion Mining," *Synthesis Lectures on Human Language Technologies, vol. 5, no. 1*, pp. 1-167, May 2012.

[4] W. Jin and H.H. Ho, "A Novel Lexicalized HMM- Based Learning Framework for Web Opinion Mining," *Proc. 26th Ann. Int'l Conf. Machine Learning*, pp.465-472, 2009.

[5] N. Jakob and I. Gurevych, " Extracting Opinion Targets in a Single and Cross- Domain Setting with Conditional Random Fields," *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 1035-1045, 2010.

[6] S. M. Kim and E. Hovy, " Extracting Opinions, Opinion Holders, and Topics Ex- pressed in Online News Media Text," *Proc. ACL/COLING Workshop Sentiment and Subjectivity in Text*, 2006.

[7] B. Pang, L. Lee, and S. Vaithyanathan, "Thumbs up?: Sentiment Classification Using Machine Learning Techniques," *Proc. Conf. Empirical Methods in Natural Language Processing*, pp. 79-86, 200.

[8] Z. Hai, K. Chang, Q. Song, and J.-J. Kim, "A Statistical NLP Approach for Feature and Sentiment Identification from Chinese Reviews," *Proc. CIPS-SIGHAN Joint Conf. Chinese Language Processing*, pp. 105-112, 2010.

[9] Priyanka U Chavan, P M Yawalkar and D V Patil. Article: "A Hybrid Approach for Recommendation System in Web Graph Mining," *International Journal of Computer Applications 95(24):23-27*, June 2014

[10] Editor, cnet. Apple iPhone6. posted to http://www.cnet.com/products/apple-iphone-6/

[11] "Zipf Law and Heap Law", (2012, Oct 13) Retrived from http://www.ccs.neu.edu

