Optimizing Cross Domain Sentiment Analysis Using Hidden Markov Continual Progression

P. Manivannan[1], C. S. Kanimozhi Selvi[2]
[1] Department of CSE, Arulmurugan College of Engineering, India
[2] Department of CSE, Kongu Engineering College, India
gotomanivannan@yahoo.co.in, kanimozhi@kongu.ac.in

Abstract

With the rapid increase in internet users and customer reviews playing major role in social media gave rise to sentiment analysis. Pre-processing of input text during sentiment analysis eliminates incomplete and noisy data. Typically, sentiment is manifested separately and applying pre-processing model for optimizing cross-domain sentiment classification is highly required. In this paper, a method called Hidden Markov Continual Progression Cosine Similar (HM-CPCS) is proposed to explore the impact of pre-processing and optimize sentiment analysis. First, a measure of subsequent and antecedent probabilities of tags is made using HM-POS Tagger for the given input dataset. Subsequent and antecedent probabilities of tags are obtained by measuring the transition probabilities between states and observations ensuring feature extraction accuracy. Next, the Continual Progression Stemmer continuously stems the text by adding prefix and suffix to form structured words for the given shortcuts and therefore reduce Error Rate Relative to Truncation (ERRT). Finally a Cosine Similarity function is applied to remove stop word for cross-domain sentiment analysis and classification. Experimental analysis shows that HM-CPCS method is able to reduce the time to extract the opinions from reviewers by 46% and improve the accuracy by 9% compared to the state-of-the-art works.

Keywords: Cross-domain, Sentiment-Classification, Hidden-Markov, Continual-Progression

1 Introduction

Reviews on broad commodity types are extensively made available on the Web like, books, hotels, movies, automobiles, restaurants and so on. The reviews available on the internet are of use to both the consumers. With the reviews extracted, sentiment analysis is being performed by several researchers and different methods have been proposed in this area.

In [1], cross-domain sentiment classification for positive or negative sentiment was made using sentiment sensitive thesaurus resulting in the accurate capturing of words that expressed similar sentiments. Helpfulness and economic impact of product reviews were analyzed in [2] using random forest based classifiers that classified review-related features, review subjectivity features and review readability features.

Despite the success of sentiment classification and analysis, sentiments of text were ignored. In [3], to learn sentiment embeddings, without feature engineering, sentiment embedding was applied to word-level sentiment analysis and therefore improving the sentiment classification performance. A platform to provide event description associated to media was designed in [4] that linked data in an efficient manner. In [5], a joint optimization model was designed to extract features appearing in both source and target, analyze label constraints and measure geometric properties ensuring sentiment classification accuracy.

High quality and personalized recommendations are the key for efficient sentiment analysis. In [6], a novel multiple rating prediction scheme was designed aiming at improving the feasibility of feature being extracted. Another novel method based on partially supervised alignment model to decrease the probability of error generation was designed in [7] with the aid of graph-based co-ranking algorithm. A computational research for mining user opinion was presented in [8].

The internet is good source for directional text that consists of text including opinions and emotions. On other hand, the web provides text-based related to consumer preferences stored in web forums, blogs, etc. Hence, sentiment analysis has evolved as a model for opinion mining from such text archives. In [9], a rule-based text feature selection method called, Feature Relation Network was designed to improve classification accuracy irrespective of feature subset size. In [10] a review of mining opinions from unstructured components was presented. A measure of domain relevance based on intrinsic and extrinsic features were made in [11] across two corpora resulting in the identification of more relevant opinion features. Candidate opinion features are taken from domain review corpus with the aid of describing a set

*Corresponding Author: P. Manivannan; E-mail: gotomanivannan@yahoo.co.in
DOI: 10.3966/160792642019052003011

of syntactic dependence rules A survey on sentiment analysis was presented in [12].

Based on aforementioned techniques and methods presented, in this work we present a method for pre-processing of sentiments in the form of shortcuts into structured words for sentiment classification. We focus on POS, stemming and stop word removal for sentiment classification.

HM-CPCS method is used to calculate the probability value for removing the noise. Followed by this, progression stemmer to stem the works for efficient conversion of shortcuts into structured words is presented using the CPS algorithm. Finally, to remove stop words present in the review, with purview of reducing the dimensionality of dataset, Cosine Similar function is applied. From our experimental results, it can be verified that the use of our HM-CPCS method provides an efficient means for pre-processing of the text documents.

The rest of the paper is structured as follows. First, Section 2 discusses related work and provides the theoretical framework for designing the objectives of our work. Then, in Section 3, the proposed method HM-CPCS is described in detail. In Section 4, the experimental results are provided followed by which in Section 5, discussion is made. Finally, Section 6 concludes the paper.

2 Related Works

Analysis of social media has resulted in tremendous potential to understand the opinion of public on wide variety of interests. In [13], the author mined Twitter to analyse and understand the public's idea of the IoT through topic modelling. Sentiment analysis is employed to enhance insights of the public's outlook towards the IoT. Pearson Correlation and Granger Casuality were analyzed in [14] to analyze the effects of twitter sentiment on stock returns. In [15], a review of feature extraction in sentiment analysis was studied.

Several changes took place in Internet with respect to economic, social, political, cultural and philosophical relations. These updations are still open, and continue to persist as Internet itself redefines its scope and reach. In [18], the correlation of sentiment score with the client assigned score value were analysed to enrich customer satisfaction assessment. A framework called Concept Level Sentiment Analysis was designed in [19] with the aid of NLP to improve extraction accuracy of the sentiments.

A prediction model based on the generalization technique based on temporal sentiment to reduce the Mean Absolute Error and Root Mean Square Error was presented in [16]. In [17], a deterministic approach for an aspect-based opinion mining was designed with the aid of NLP-based rules to improve the accuracy and recall rate for tourism product reviews was presented.

However, due to shortcuts and unstructured words used in blogs, twitters and several social sites, users are often overwhelmed with information when trying to analyze various reviews. So far, many authors have tacked the problem of providing meaningful analysis for the unstructured words from large number of reviews relying on data-mining-based tools. Considering a similar problem, this work is an effort to provide a pre-processing method to convert shortcuts used in twitters, blogs into meaningful words and help users digest in an easy manner the vast availability of sentiments.

3 Methodology

In this section, pre-processing of online reviews extracted from Sentiwordnet is performed to remove the noise present in it before the processing of sentiment mining. This is because on line reviews consists of short informal texts and contain mistakes in spelling, inclusion of words not present in the dictionary, punctual mistakes, capitalization error and so on.

As shown in the Figure 1, the block diagram of HM-CPCS comprises of three components. The first, is the input dataset where a lexical source for opinion mining called Sentiwordnet is used. The second, involves pre-processing that performs POS tagging, Continual Progression Stemmer (CPS) and Cosine Similar Stop Word Removal respectively. The HM-CPCS works on the principle of optimization of dimensionality reduction where the short informal texts are converted to structured words in an optimal manner, resulting in the accuracy of features being selected.

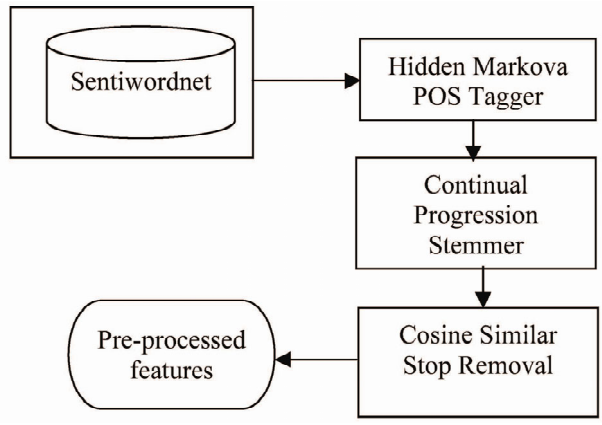

Figure 1. Block diagram of HM-CPCS

3.1 Hidden Markov POS Tagger

In this section, a HM-POS Tagger is discussed that assigns the best tag to a word by measuring subsequent and antecedent probabilities of tags to the Sentiwordnet, a lexical resource for word polarity provided as an input.

As shown in the Figure 2, the HM-POS Tagger consists of states 's_i=s_1, s_2, … s_n', observations 'o_i=o_1,

o_2, … o_n', transition probabilities 'u_i=u_1, u_2, … u_n' and output probabilities 'v_i=v_1, v_2, … v_n' and tags 'T_1, T_2'. Here, states represent the domains (i.e. document 'D_i') whereas the observations are the representation for tags that includes the review statement consisting of words 'W_i=W_1, W_2, … W_n' with tags 'T_i=T_1, T_2, … T_n'. Then, the probability function is expressed as given below.

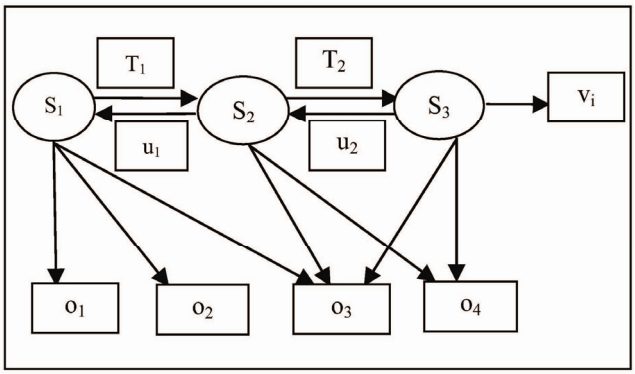

Figure 2. Structure of HM-POS tagger

$$Prob(T_i \mid W_i) = Prob(T_i \mid T_{i-1}) * Prob(T_{i+1} \mid T_i) * Prob(W_i \mid T_i) \quad \textbf{(1)}$$

From (1), '(T_i|T_{i-1})' refers the probability of subsequent tag given the antecedent tag, '(T_{i+1}|T_i)' refers the probability of succeeding tag given the subsequent tag. From (1), the transition between the tags is captured and is expressed as given below.

$$Prob(T_i \mid T_{i-1}) = \frac{count(T_{i-1} \mid T_i)}{count(T_{i-1})} \quad \textbf{(2)}$$

Each tag transition probability is obtained by measuring the ratio of count of two tags seen together in a domain 'Count(T_{i-1}, T_i)' to the count of subsequent tag seen independently 'Count(T_{i-1})' in a domain. This is performed because occurrence of certain tags is preceded by the occurrence of certain other tags. Therefore, using Hidden Markov, the probability 'Prob(Adj|N)' fetches higher score when compared to the probability 'Prob(Adj|PN)', where 'Adj', 'N' and 'PN' refers to adjectives, nouns and pronouns respectively. Similarly, the word occurrence probability is expressed as given below.

$$Prob(W_i \mid T_i) = \frac{count(T_v W_i)}{count(T_i)} \quad \textbf{(3)}$$

From (3), the word occurrence probability is the ratio of count of tag to be measured and the word occurring together in the domain 'Count(T_i, W_i)' divided by the count of tag in a domain 'Count(T_i)'. Figure 3 shows the algorithmic description of HM-POS in sentiment analysis.

Input: Document 'D_i', Reviews 'R_i', Words '$W_i = W_1, W_2, ..., W_n$', Tags '$T_i = T_1, T_2, ..., T_n$'
Output: Improved rate of accuracy
1. Begin
2. For each Document 'D_i' with Reviews 'R_i'
3. Measure transition between tags using (2)
4. Measure word occurrence probability using (3)
5. End for
6. End

Figure 3. HM-POS algorithm

As shown in the figure, for each document and reviews extracted from Sentiwordnet, the HM-POS algorithm assigns a tag to each word in the text document. With the input obtained from Sentiwordnet, a lexical database, where subsequent and antecedent tags are used to measure the transition and word occurrence probability between tags. This in turn ensures feature extraction accuracy.

3.2 Continual Progression Stemmer

In this section, the process of stemming is presented using CPS that converts all shortcuts to structured words with the support of HM-POS Tagger. For example, 'automat' is converted to either 'automatic,' 'automate,' or 'automation' using CPS. Followed by this applying HM-POS algorithm to the resultant stem that uses probability of occurrence of words and transition between tags identifies, where automatic, automate or automation best fits. Motivated by the work of [4] porter stemming, where suffix were replaced by the prefix, the HM-CPCS replaces prefix with suffix. The formulation is as given below.

From (4), the prefix 'FUL' is replaced by the suffix 'FULNESS', if and only if the stem factor is greater than zero 'S > 0'. This is performed as a Continual Progression model, hence called as CPS. Figure 4 shows the activity diagram of CPS.

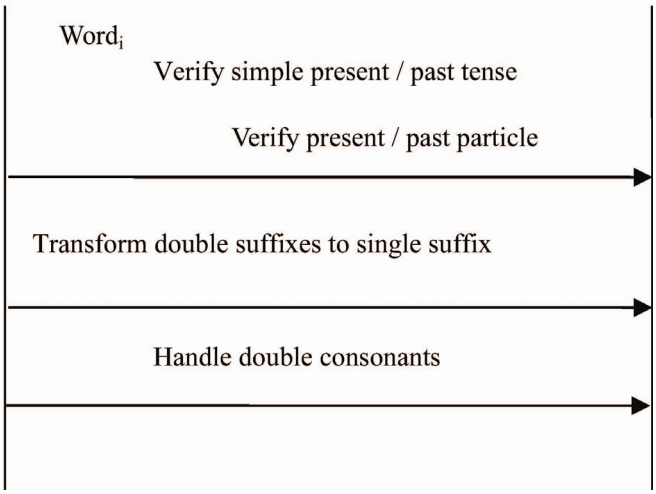

Figure 4. Activity diagram of CPS

With the help of the activity diagram of CPS, the CPS algorithm is constructed in the proposed work that consists of three steps. Each step explains a set of rules to be checked for. In order to stem a word, the HM-CPCS tests the rules in a continual progression manner. Followed by which, the rules are checked and according to the satisfaction of the rules, the corresponding action takes place. This process is continued until all the words are stemmed. Figure 5 given below explains the CPS algorithm.

Input: Words 'W_i=W_1, W_2, … W_n', Rules '$Rule_i$=$Rule_1$, $Rule_2$, $Rule_3$'
Output: Reduces ERRT
1. **Begin**
2. Assign Rule [1] = Handles plurals, present tense, past participle
3. Assign Rule [2] = Transform double suffixes to single suffix
4. Assign Rule [3] = Handles double consonant
5. **Repeat**
6. **For** each Word 'W_i'
7. **If** Rule [i] = 'W_i'
8. Execute Rule [i]
9. **Else If** Rule [i] <> 'W_i'
10. Go to 6
11. **End if**
12. **End for**
13. **Until** (all words are processed)
14. **End**

Figure 5. CPS Algorithm

As shown in Figure 5, the CPS algorithm assigns three rules, where the first rules handles simple present/past tense and simple present/past participle. For example words ending with 'i' are transformed to 'y' (i.e. happi to happy). The second rule transforms single suffix to double suffixes (i.e. oscillate to oscillator or oscillation). Finally, the third rule handles double consonants (i.e. attribut to attribute, attributes). In the CPS algorithm, with the arrival of each word, the three rules are tested in a continual progression manner. If one of the rules matches the word, then the condition attached to that rule gets executed, otherwise, checks and tests for other rules. This process is repeated for all words in the Sentiwordnet. By measuring the occurrence of words and transition between tags, appropriate structured words are replaced with the shortcut words. This in turn reduces the ERRT.

3.3 Cosine Similar Stop Word Removal

Stop words are a division of natural language. The motive that stop-words are eliminated from a text because they build the text looks heavier and less significant for analysts. Removing stop words minimizes the dimensionality of term space. The most general words in text documents are articles, prepositions, and pro-nouns, etc in which they unable to give the meaning of the documents. These words are treated like stop words. Example for stop words include 'the, in, a, an, with', etc.

In this section, the removal of stop words using Cosine Similarity function is presented. High frequency words like 'a', 'an', 'the', 'of' that occurs frequently are called as common words or stop words that increases the dimensionality of dataset and therefore increases the preprocessing time. Different methods are available for stop-word elimination [4, 15]. To reduce the dimensionality of dataset and preprocessing time, Cosine Similar Stop Word Removal is applied in HM-CPCS.

The Cosine Similar Stop Word Removal in HM-CPCS measures word intensity 'WI' that evaluates the intensity of a word for identifying two cross-domains. The two cross-domains in HM-CPCS are selected in such a way that they are more similar. For example, for two cross-domains, 'D_i' and 'D_j', the word intensity 'WI', the term intensity ' I(W_i)' of word 'W_i' is expressed in terms of the following probability.

$$WI(W_i) = Prob(W_i \in D_i \mid W_i \in D_j) \quad \textbf{(5)}$$

With the word intensity measured using (5), the similarity of two cross-domains in HM-CPCS is evaluated using the Cosine Similar function.

$$Sim(W_i) = Cos(\theta) = \frac{D_i D_j}{|D_i \| D_j|} = \frac{\sum_{i,j=1}^{n} D_i D_j}{\sum_{i=1}^{n} D_i^2 \sqrt{\sum_{j}^{n} {}_1 D_j^2}} \quad \textbf{(6)}$$

From (6), the first domain of pair is selected in random manner. In order to find similarity, word intensity of domain D_i is compared to word intensity of domain D_j. The HM-CPCS assigns a similarity threshold factor to be 'STF'. The cosine similarity function of a word is then compared to 'WI' and if the word intensity is found greater than 'STF', then the cross domains are said to be related otherwise, not related. Figure 6 shows Cosine Similar Cross-domain algorithm.

Input: Cross-domains, 'D_i' and 'D_j', Word 'W_i=W_1, W_2, … W_n', Similarity Threshold Factor 'STF'
Output: Optimizes pre-processing
1. **Begin**
2. Assign Similarity Threshold Factor 'STF'
3. **For** each Cross-domains, 'D_i'
4. **For** each, Word 'W_i'
5. Measure word intensity 'WI' using (5)
6. Measure similarity of two cross-domains with Cosine Similar function using (6)
7. **End for**
8. **End for**
9. **End**

Figure 6. Cosine similar cross domain algorithm

As shown in the Figure 6, for each cross-domains and words in it, the Cosine Similar Cross-domain algorithm measure the word intensity which evaluates how informative a word is for identifying from one or more domains. Then next issue handled in the Cosine Similar Cross-domain algorithm is the measure of relatedness of two cross-domains performed using Cosine Similarity function.

4 Experimental Settings

Proposed HM-CPCS is experimented using standard benchmark data sets of consumer product and services reviews extracted from Sentiwordnet data set and OpinRank Review data set. It includes user reviews of cars and hotels collected from Tripadvisor (~259,000 reviews) and Edmunds (~42,230 reviews). To evaluate the performance of HM-CPCS, SentiWordNet [1] a lexical resource in which each WordNet synset is associated with a polarity score is used to accurately predict the polarity of words. SentiWordNet assigns each synset in WordNet3 that pre-processes the text present in it. For the evaluation of HM-CPCS, the dataset is randomly divided into 60% training and 40% testing documents, so that both training and testing dataset are disjoint. The experiments in HM-CPCS are repeated for 10 times, and final performance is reported by averaging the results.

Accuracy is used as an evaluation measure. It is computed by the total number of correct pre-processed reviews to the total number of reviews in target domain. The feature pre-processing accuracy is a measure to evaluate the significance of optimization. The accuracy is measured using the ratio of number of correct pre-processed reviews to the total number of reviews in target domain is given below.

$$A = \sum_{i=1}^{n} \frac{correct_{pr}}{D_i R_i} \quad \textbf{(7)}$$

From (7), accuracy factor 'A' is measured using total reviews 'R_i' in domain 'D_i' to the correct pre-processed reviews '$Correct_{pr}$' respectively. It is measured in terms of percentage. The second evaluation measure used in the HM-CPCS is ERRT. The ERRT is a useful measured for deciding on the best overall stemmer in cases where one stemmer is better in terms of under-stemming but worse in terms of over-stemming. The ERRT that measures the words (i.e. stemmers) incorrectly identified as belonging to a specific review in target domain to the total number of reviews. The ERRT is mathematically formulated as given below.

$$ERRT = \sum_{i=1}^{n} \frac{SR_i D_i}{R_i D_i} \quad \textbf{(8)}$$

From (8), the 'ERRT' is obtained using the reviews in domain '$R_i D_i$'. It is measured in terms of percentage. Lower the ERRT, more efficient the method is said to be. Final and the third evaluation measure is the time taken for removing the stop words. The execution time (ET) for stop word removal is the time taken to remove the stop words with respect to the number of review words provided in a domain. The execution time for stop word removal is mathematically formulated as given below.

$$ET = \sum_{i=1}^{n} R_i * Time\,(Sim(W_i)) \quad \textbf{(9)}$$

From (9), the 'ET' is obtained using the review words 'R_i' and time for stop word removal based on the word similarity '$Sim(W_i)$' and is measured in terms of milliseconds.

5 Discussion

The performance of HM-CPCS is compared with the existing Sentiment Sensitive Thesaurus (SST) [1] and Mining Text and Reviewer Characteristics (MTRC) [2]. The performance is evaluated according to the following metrics.

5.1 Impact of Accuracy

To better understand the effectiveness of the proposed HM-CPCS method, extensive experimental results are reported in Table 1 and comparison is made with two other methods SST and MTRC using Java language. The accuracy in Table 2 is observed to increase with 20 customer review words. With the increase in the review words, the accuracy remains stable. In order to observe the accuracy for achieving optimal sentiment classification, a scenario with default parameters value for seven different periods was measured at different time intervals. For each implementation run, the review words obtained from each customer (i.e. tourists) was changed.

Table 1. Tabulation for accuracy

Number of Reviews	Accuracy (%)					
	Sentiwordnet Dataset			OpinRank Reviewer Dataset		
	HM CPCS	SST	MTRC	HM CPCS	SST	MTRC
10	8.82	14.32	15.32	79.25	73.28	67.34
20	14.31	22.15	26.78	85.41	79.31	69.31
30	17.35	31.28	33.35	88.62	83.75	75.22
40	24.32	39.41	42.90	92.46	86.33	78.31
50	33.98	48.90	57.67	83.15	77.46	70.34
60	41.32	59.32	78.32	86.32	80.02	74.05
70	49.35	67.87	89.32	90.11	85.91	79.51

Table 2. Tabulation for ERRT

Number of Reviews	Accuracy (%)					
	Sentiwordnet Dataset			OpinRank Reviewer Dataset		
	HM CPCS	SST	MTRC	HM CPCS	SST	MTRC
10	8.82	14.32	15.32	79.25	73.28	67.34
20	14.31	22.15	26.78	85.41	79.31	69.31
30	17.35	31.28	33.35	88.62	83.75	75.22
40	24.32	39.41	42.90	92.46	86.33	78.31
50	33.98	48.90	57.67	83.15	77.46	70.34
60	41.32	59.32	78.32	86.32	80.02	74.05
70	49.35	67.87	89.32	90.11	85.91	79.51

Results are presented for different number of reviews where the Sentiwordnet is divided into four categories, adjective, adverb, verb and noun. The results reported here confirm that with increase in the number of reviews, the accuracy factor gets increased but not observed to be linear because of noise. Comparatively, the accuracy is improved using HM-CPCS. The accuracy value gets saturated when the number of opinion words ranges reaches 40.

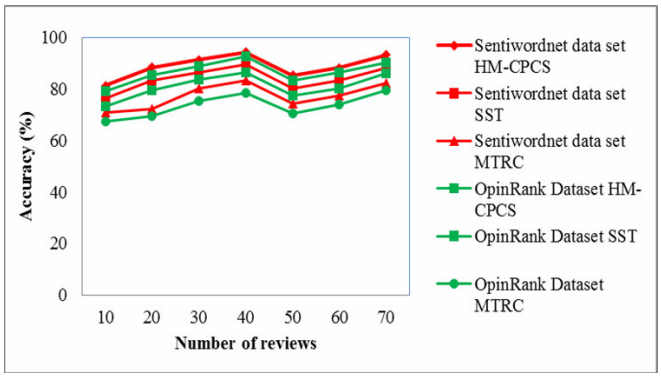

Figure 7. Measure of accuracy with respect to number of reviews

However, comparative graph shows that HM-CPCS method to illustrate better performance achievement than the counterparts [1-2]. This is because of the application of HM-POS Tagger. Using the Probability functions in HM-POS Tagger, the word intensity is measured which forms the basis for evaluating the transition between tags. As a result, the accuracy on customer review words is seen to be improved using Sentiwordnet dataset in HM-CPCS method by 6% compared to SST and 13% compared to MTRC respectively. In addition, Opinrank reviewer dataset improves the accuracy by 7% and 18% using HM-CPCS method when compared to SST and MTRC respectively.

5.2 Impact of Error Rate Relative to Truncation

In order to reduce the ERRT, HM-CPCS method is used that reduces the ERRT rate. With thousands of reviews in Sentiwordnet, the training dataset includes number of reviews in the range of 10 to 70. The results of 7 different opinion reviews obtained from Sentiwordnet for experimental setup are listed in Table 2.

Figure 8 shows the behavior of the ERRT in response to varying number of reviews made from different domains comprising of adverb, verb, adjective and noun. The average ERRT of the three methods was observed to be increasing with the number of reviews made in the range of 10 and 70. There was a fall off in the values of ERRT when 40 reviews were made and then a rise in the ERRT was observed. This is because of the involvement of high variations observed in the reviews obtained from different domains, a steadiness was not observed and ERRT varied accordingly

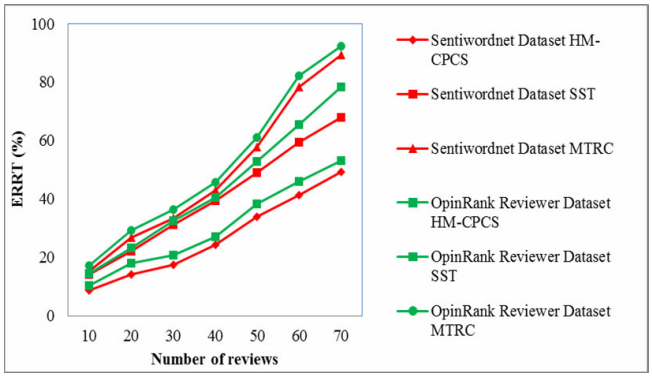

Figure 8. Measure ERRT with respect to different number of reviews

Comparatively, the HM-CPCS observed a decreased ERRT when compared to SST and MTRC. This is because the HM-CPCS not only applies CPS for stemming but also applies HM-POS tagger that measures probability of occurrence of words and transition between tags to handle different review words that results in the decrease in the ERRT by 61% compared to SST Sentiwordnet dataset. Besides, using the HM-CPCS the rules are tested in a continual progression manner using CPS algorithm forming a decrease in the ERRT by 89% compared to MTRC. With the aid of using Opinrank reviewer dataset, proposed HM-CPCS method reduces the ERRT by 30% and 41% when compared to SST and MTRC respectively.

5.3 Impact of Execution time for Stop Word Removal

To assess the performance of execution time for stop word removal with respect to different reviews using the three methods HM-CPCS, SST and MTRC is provided in Table 3.

Table 3. Tabulation for execution time

Number of Reviews	Accuracy (%)					
	Sentiwordnet Dataset			OpinRank Reviewer Dataset		
	HM CPCS	SST	MTRC	HM CPCS	SST	MTRC
10	15.29	27.32	29.35	20.46	28.33	32.15
20	20.34	32.14	35.90	24.38	34.05	37.94
30	24.18	36.89	41.32	28.64	39.45	44.23
40	33.21	45.78	50.38	37.22	48.19	53.24
50	40.35	52.14	54.28	44.15	53.05	57.16
60	49.09	52.67	55.78	51.06	54.34	59.33
70	57.89	69.98	72.32	60.21	70.53	75.61

Here, each review consists of number of words extracted from different domains of various user reviews measured in terms of percentage. The results on HM-CPCS method are investigated with small stage information which is obtained from experimental work.

Figure 9 indicate execution time for HM-CPCS is lesser than SST and MTRC. This is because of the application of Cosine Similar Function stop word removal where the removal of stop words is made according to the word intensity and similarity of two cross-domains with the aid of Cosine Similar function. This in turn confirms the reduced execution time for stop word removal by applying HM-CPCS than SST and MTRC.

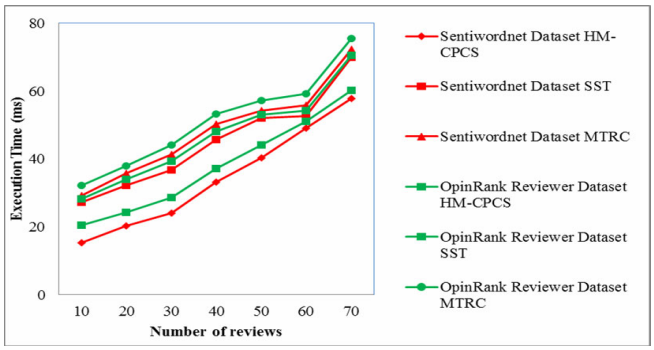

Figure 9. Measure of execution time with respect to number of reviews

Another interesting observation from Figure 9 is that by applying Cosine Similar Cross-domain algorithm in HM-CPCS it was capable of evaluating word intensity from overall reviews by updating the words in cross-domains and a measure of relatedness of two cross-domains was performed using Cosine Similarity function. Baseline results were lower than 20.34ms when 10-20 reviews were considered, indicating that the error rate during initial selection of the reviews were considerably less. On the other hand, the upper-bound with increasing number of reviews with different domains saw a good result using Sentiwordnet dataset by reducing the execution time by 41% compared to SST and 51% compared to MTRC respectively. Similarly, Opinrank reviewer dataset reduces the execution time by 21% and 28% in proposed HM-CPCS method when compared to SST and MTRC respectively.

6 Conclusion

A Cross-domain Sentiment analysis using Hidden Markov Continual Progression Cosine Similar Function is presented. To perform an efficient pre-processing method in cross-domain sentiment analysis, use POS tagger based on the Hidden Markov with reviews collected from different domains and facilitates feature extraction. Then, a stemmer model using Continual Progression is designed to expand reviews during train and test times to measure transition probabilities of the review words based on the provided observations. Finally, similarity function is used to measure the relatedness of cross-domains using cosine factor. The experimental results evidences that the HM-CPCS is better in terms of both the accuracy and the computational performance.

References

[1] D. Bollegala, D. Weir, J. Carroll, Cross-Domain Sentiment Classification Using a Sentiment Sensitive Thesaurus, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 25, No. 8, pp. 1719-1731, August, 2013.

[2] A. Ghose, P. G. Ipeirotis, Estimating the Helpfulness and Economic Impact of Product Reviews: Mining Text and Reviewer Characteristics, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, No. 10, pp. 1498-1512, October, 2011.

[3] D. Tang, F. Wei, B. Qin, N. Yang, T. Liu, M. Zhou, Sentiment Embeddings with Applications to Sentiment Analysis, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 2, pp. 496-509, February, 2016.

[4] H. Khrouf, V. Milicic, R. Troncy, Mining Events Connections on the Social Web: Real-time Instance Matching and Data Analysis in EventMedia, *Web Semantics:* Science, *Services and Agents on the World Wide Web*, Vol. 24, pp. 3-10, January, 2014.

[5] D. Bollegala, T. Mu, J. Y. Goulermas, Cross-domain Sentiment Classification Using Sentiment Sensitive Embeddings, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 28, No. 2, pp. 398-410, February, 2016.

[6] W. Jiang, J. Wu, G. Wang, H. Zheng, Forming Opinions via Trusted Friends:Time-evolving Rating Prediction Using Fluid Dynamics, *IEEE Transactions on Computers*, Vol. 65, Issue 4, pp. 1211-1224, April, 2016.

[7] K. Liu, L. Xu, J. Zhao, Co-Extracting Opinion Targets and Opinion Words from Online Reviews Based on the Word Alignment Model, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 27, No. 3, pp. 636-650, March, 2015.

[8] G. Petz, M. Karpowicz, H. Fürschuß, A. Auinger, V. Stříteský, A. Holzinger, Computational Approaches for Mining User's Opinions on the Web 2.0, *Information*

Processing & Management, Vol. 50, No. 6, pp. 899-908, November, 2014.

[9] A. Abbasi, S. France, Z. Zhang, H. Chen, Selecting Attributes for Sentiment Classification Using Feature Relation Networks, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 23, Issue 3, pp. 447-462, March, 2011.

[10] K. Khan, B. Baharudin, A. Khan, A. Ullah, Mining Opinion Components from Unstructured Reviews: A Review, Elsevier, *Journal of King Saud University - Computer and Information Sciences*, Vol. 26, No. 3, pp. 258-275, September, 2014.

[11] Z. Hai, K. Chang, J.-J. Kim, C. C. Yang, Identifying Features in Opinion Mining via Intrinsic and Extrinsic Domain Relevance, *IEEE Transactions on Knowledge and Data Engineering*, Vol. 26, No. 3, pp. 623-634, March, 2014.

[12] D. M. E.-D. M. Hussein, A Survey on Sentiment Analysis Challenges, *Journal of King Saud University - Engineering Sciences*, pp. 1-9, April, 2016.

[13] J. Bian, K. Yoshigoe, A. Hicks, J. Yuan, Z. He, M. Xie, Y. Guo, M. Prosperi, R. Salloum, F. Modave, Mining Twitter to Assess the Public Perception of the Internet of Things, *PLoS ONE*, Vol. 11, No. 7, pp. 1-14, July, 2016.

[14] G. Ranco, D. Aleksovski, G. Caldarelli, M. Grčar, I. Mozetič, The Effects of Twitter Sentiment on Stock Price Returns, *PLoS ONE*, Vol. 10, No. 9, pp. 1-21, September, 2015.

[15] M. Zubair Asghar, A. Khan, S. Ahmad1, F. M. Kundi, A Review of Feature Extraction in Sentiment Analysis, *Journal of Basic Appl. Sci. Res.*, Vol. 4, No. 3, pp. 181-186, February, 2014.

[16] P. G. Preethia, V. Umab, Ajit kumarc, Temporal Sentiment Analysis and Causal Rules Extraction from Tweets for Event Prediction, Elsevier, *Procedia Computer Science*, Vol. 48, pp. 84-89, May, 2015.

[17] E. Marrese-Taylor, J. D. Velásquez, F. Bravo-Marquez, A Novel Deterministic Approach for Aspect-based Opinion Mining in Tourism Products Reviews, Elsevier, *Expert Systems with Applications*, Vol. 41, No. 17, pp. 7764-7775, December, 2014.

[18] M. D. Miranda, R. J. Sassi, Using Sentiment Analysis to Assess Customer Satisfaction in an Online Job Search Company, Springer, *Business Information Systems Workshops*, Vol. 183 of the Series Lecture Notes in Business Information Processing, pp. 17-27, October, 2014.

[19] E. Cambria, S. Poria, F. Bisio, R. Bajpai, I. Chaturvedi, The CLSA Model:A Novel Framework for Concept-Level Sentiment Analysis, *Computational Linguistics and Intelligent Text Processing*, Vol. 9042, pp. 3-22, April, 2015.

Biographies

P. Manivannan is a faculty member in the Department of CSE of Arulmurugan College of Engineering, India. Currently doing Ph.D. in Anna University, India. He published 2 articles in international journals.

C. S. Kanimozhi Selvi is a faculty member in the Department of CSE of Kongu Engineering College, India. She completed Ph.D in 2011. She has published 20 articles in international journals and 30 papers in international conferences.