# Automatic retrieval and clustering of similar words

**Priya S**
ME computer science and Engineering, Infant Jesus College of Engineering and Technology
priyastalin90@gmail.com
**Mr. A. Jegadeesh, M.E, (Ph.D)**
Assistant Professor, Dept of computer science and Engineering,
Infant Jesus College of Engineering and Technology

***Abstract*—Automatic classification of sentiment is important for numerous applications such as opinion mining, opinion summarization, contextual advertising, and market analysis. Typically, sentiment classification has been modeled as the problem of training a binary classifier using reviews annotated for positive or negative sentiment. However, sentiment is expressed differently in different domains, and annotating corpora for every possible domain of interest is costly. Applying a sentiment classifier trained using labeled data for a particular domain to classify sentiment of user reviews on a different domain often results in poor performance because words that occur in the train (source) domain might not appear in the test (target) domain. We propose a method to overcome this problem in cross-domain sentiment classification. First, we create a sentiment sensitive distributional thesaurus using labeled data for the source domains and unlabeled data for both source and target domains. Sentiment sensitivity is achieved in the thesaurus by incorporating document level sentiment labels in the context vectors used as the basis for measuring the distributional similarity between words. Next, we use the created thesaurus to expand feature vectors during train and test times in a binary classifier. The proposed method significantly outperforms numerous baselines and returns results that are comparable with previously proposed cross-domain sentiment classification methods on a benchmark data set containing Amazon user reviews for different types of products. We conduct an extensive empirical analysis of the proposed method on single- and multisource domain adaptation, unsupervised and supervised domain adaptation, and numerous similarity measures for creating the sentiment sensitive thesaurus. Moreover, our comparisons against the SentiWordNet, a lexical resource for word polarity, show that the created sentiment-sensitive thesaurus accurately captures words that express similar sentiments.**

***Index Terms*— Cross-domain sentiment classification, domain adaptation, thesauri creation**

## I. INTRODUCTION

USERS express their opinions about products or services they consume in blog posts, shopping sites, or review sites. Reviews on a wide variety of commodities are available on the Web such as, books (amazon.com), hotels (tripadvisor.com), movies (imdb.com), automobiles (caranddriver.com -), and restaurants (yelp.com). It is useful for both the consumers as well as for the producers to know what general public think about a particular product or service. Automatic document level sentiment classification is the task of classifying a given review with respect to the sentiment expressed by the author of the review. For example, a sentiment classifier might

classify a user review about a movie as positive or negative depending on the sentiment expressed in the review. Sentiment classification has been applied in numerous tasks such as opinion mining opinion summarization contextual advertising and market analysis. For example, in an opinion summarization system it is useful to first classify all reviews into positive or negative sentiments and then create a summary for each sentiment type for a particular product. A contextual advert placement system might decide to display an advert for a particular product if a positive sentiment is expressed in a blog post.

Supervised learning algorithms that require labeled data have been successfully used to build sentiment classifiers for a given domain. However, sentiment is expressed differently in different domains, and it is costly to annotate data for each new domain in which we would like to apply a sentiment classifier. For example, in the electronics domain the words “durable” and “light” are used to express positive sentiment, whereas “expensive” and “short battery life” often indicate negative sentiment. On the other hand, if we consider the books domain the words “exciting” and “thriller” express positive sentiment, whereas the words “boring” and “lengthy” usually express negative sentiment. A classifier trained on one domain might not perform well on a different domain because it fails to learn the sentiment of the unseen words.

## 2. LITERATURE SURVEY

Sentiment classification using machine learning techniques The effort to better organize this information for users, researchers have been actively investigating the problem of automatic text categorization. A challenging asp ect of this problem that seems to distinguish it from traditional topic-based classification is that while topics are of-ten identifiable by keywords alone, sentiment can be expressed in a more subtle manner. Most previous research on sentiment-based classification has been at least partially knowledge-based. Some of this work focuses on classifying the semantic orientation of individual words or phrases, using linguistic heuristics or a pre-selected set of seed words. This domain is experimentally convenient because there are large on-line collections of such re-views, and because riveters often summarize their overall sentiment with a machine-extractable rating indicator. But the stress that the machine learning methods and features the use are not septic to movie reviews, and should b e easily applicable to other domains as long as sufficient training data exists. Ratings there automatically extracted and converted into one of three categories: positive, negative, or neutral.

An expert on using ma-chine learning for text categorization predicted relatively low performance for automatic methods. On the other hand, it seems that distinguishing positive from negative reviews is relatively easy for humans, especially in comparison to the standard text categorization problem, where topics can b e closely related. The brevity of the human-produced lists is a factor in the relatively poor performance results, it is not the case that size alone necessarily limits accuracy. Naive Bayes classification, maximum entropy classification, and sup-p ort vector machines. The philosophies behind these three algorithms are quite different, but each has been shown to be effective in previous text categorization studies. In addition to looking specifically for negation words in the context of a word, the also studied the use of bigrams to capture more context in general. This would not rule out the possibility that bigram presence is as equally useful a

feature as unigram presence. Unigram presence information turned out to b e the most effective; in fact, none of the alternative features the employed provided consistently better performance once unigram presence was incorporated. Interestingly, though, the superiority of presence information in comparison to frequency information in our setting contradicts previous observations made n topic-classification work.

## 3. OVERVIEW

### 3.1. EXISTING SYSTEM

Very large amounts of information are available in on-line documents. To better organize this information for users, researchers have been investigating the problem of automatic text categorization. The bulk of such work has focused on topical categorization, attempting to sort documents according to their subject matter. Hot ever, recent years have seen rapid growth in on-line discussion groups and review sites where a crucial characteristic of the posted articles is their sentiment, or overall opinion towards the subject matter, a product review is positive or negative. Sentiment classification would also be helpful in business intelligence applications and recommender systems, where user input and feedback could be quickly summarized using No adapt and NSST. This benchmark dataset has been used in much previous work on cross-domain sentiment classification and by evaluating on it the can directly compare.

**DEMERITS:**

Can't predict the correct sentimental meaning for the document.Classifier trained on one domain might not be efficient.Classifier meanings should not be matched among another domain.In single-domain sentiment classification, a classifier is trained using labeled data annotated from the domain in which it will be applied.

### 3.2. PROPOSED SYSTEM

The propose a new technique called cross-domain sentiment classification .By using technique, Identify which source domain features are related to which target domain features. Required a learning framework to incorporate the information regarding the relatedness of source and target domain features. A fully automatic method to create a thesaurus that is sensitive to the sentiment of words expressed in different domains. Utilize a both labeled and unlabeled data available for the source domains and unlabeled data from the target domain using SSN. Lexical elements can be derived from both labeled and unlabeled reviews whereas; sentiment elements can be derived only from labeled reviews. The method to reach its full performance with a small number of source domain labeled instances is particularly important when applying to domains with a few labeled instances. The availability of unlabeled data for the construction of a sentiment sensitive thesaurus significantly out-performs several baselines and reports results that are comparable with previously proposed cross-domain sentiment classification methods on a benchmark dataset.

## 4. RELATED WORK

Sentiment classification systems can be broadly categorized into single-domain and cross domain classifiers based upon the domains from which they are trained on and subsequently applied to. On another axis, sentiment classifiers can be categorized depending on whether they classify sentiment at word level sentence level or document level. Our method performs cross-domain sentiment classification at document level.

In single-domain sentiment classification, a classifier is trained using labeled data annotated from the domain in which it will be applied. Turney measures the co occurrences between a word and a set of manually selected positive words (e.g., good, nice, excellent, and so on) and negative words (e.g., bad, nasty, poor, and so on) using point wise mutual information to compute the sentiment of a word. Kanayama and Nasukawa proposed an approach to build a domain-oriented sentiment lexicon to identify the words that express a particular sentiment in a given domain. By construction, a domain specific lexicon considers sentiment orientation of words in a particular domain. Therefore, their method cannot be readily applied to classify sentiment in a different domain.

Compared to single-domain sentiment classification, which has been studied extensively in previous work cross-domain sentiment classification has only recently received attention with the advancement in the field of domain adaptation Aue and Gammon report a number of empirical tests on domain adaptation of sentiment classifiers. They use an ensemble of nine classifiers to train a sentiment classifier. However, most of these tests were unable to outperform a simple baseline classifier that is trained using all labeled data for all domains. They acknowledge the challenges involved in cross-domain sentiment classification and suggest the possibilities of using unlabeled data to improve performance.

Blitzer proposes the SCL algorithm to train a cross-domain sentiment classifier. SCL is motivated by the alternating structural optimization (ASO), a multitask learning algorithm, proposed by Ando and Zhang Given labeled data from a source domain and unlabeled data from both source and target domains, SCL chooses a set of pivot features which occur frequently in both source and target domains. Next, linear predictors are trained to predict the occurrences of those pivot features. Positive training instances for a particular pivot feature are automatically generated by removing the corresponding pivot feature in feature vectors. Feature vectors that do not contain a particular pivot feature are considered as negative training instances for the task of learning a predictor for that pivot feature. It is noteworthy that this approach does not require any manually labeled feature vectors for learning the pivot feature predictors. For each pivot feature, a linear weight vector is computed and the set of weight vectors for all the pivot features under consideration are arranged in a matrix. Next, SVD is performed on this weight matrix to construct a lower dimensional feature space. Each feature vector is then mapped to a lower dimensional representation by multiplying with the computed matrix. Finally, each original feature vector is augmented with its lower dimensional representation to form a new (extended) feature vector. A binary classifier is trained using labeled reviews (positive and negative sentiment labels) using this new set of feature vectors. In the SCL-MI approach, a variant of the SCL approach, mutual information between a feature and the source label is used to select pivot features instead of the co occurrence frequency. However, in practice it is hard to construct a reasonable number of auxiliary tasks from data, which might limit the transfer ability of SCL for cross-domain sentiment classification. Moreover, the heuristically selected pivot features might not guarantee the best performance on target domains. In contrast, our method uses all features when creating the thesaurus and selects a subset of features during training using L1 regularization. Moreover, we do not require SVD, cubic in time complexity, which can be computationally costly for large data sets.

## 5. MODULES

The Modules are

- Loading Dataset
- POS Tagging
- Lexical Elements
- Sentiment Sensitive Element

**A. Loading Dataset**

We use the cross-domain sentiment classification data set1 prepared by Blitzer et al. to compare the proposed method against previous work on cross-domain sentiment classification. This data set consists of Amazon product reviews for four different product types: books, DVDs, electronics, and kitchen appliances. Each review is assigned with a rating (0-5 stars), a reviewer name and location, a product name, a review title and date, and the review text. Reviews with rating >3 are labeled as positive, whereas those with rating <3 are labeled as negative. The overall structure of this benchmark data set is shown in Table 3. For each domain, there are 1,000 positive and 1,000 negative examples, the same balanced composition as the polarity data set constructed by Pang The data set also contains some unlabeled reviews for the four domains. This benchmark data set has been used in much previous work on cross-domain sentiment classification and by evaluating on it we can directly compare the proposed method against existing approaches.

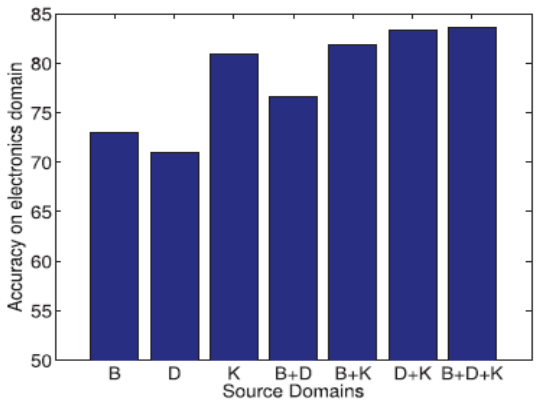
Following previous work, we randomly select 800 positive and 800 negative labeled reviews from each domain as training instances (total number of training instances are 1;600 _ 4 ¼ 6;400), and the remainder is used for testing (total number of test instances are 400 _ 4 ¼ 1;600). In our experiments, we select each domain in turn as the target domain, with one or more other domains as sources.

We use classification accuracy on target domain as the evaluation metric. It is the fraction of the correctly classified target domain reviews from the total number of reviews in the target domain, and is defined as follows:

$$\text{Accuracy} = \frac{\text{no. of correctly classified target reviews}}{\text{total no. of reviews in the target domain}}. \quad (1)$$

**B. POS Tagging**

Part-of-speech tagging (POS tagging or POST), also called grammatical tagging or word-category disambiguation, is the process of marking up a word in a text (corpus) as corresponding to a particular part of speech, based on both its definition, as thell as its context—i.e. relationship with adjacent and related words in a phrase, sentence, or paragraph. A simplified form of this is commonly taught to school-age children, in the identification of words as nouns, verbs, adjectives, adverbs, etc. Once performed by hand, POS tagging is now done in the context of computational linguistics, using algorithms which associate discrete terms, as thell as hidden parts of speech, in accordance with a set of descriptive tags. POS-tagging algorithms fall into two distinctive groups: rule-based and stochastic.



E. Brill's tagger, one of the first and widely used English POS-taggers, employs rule-based algorithms.

A Part-Of-Speech Tagger (POS Tagger) is a piece of software that reads text in some language and assigns parts of speech to each word (and other token), such as noun, verb, adjective, etc., although generally computational applications use more fine-grained POS tags like 'noun-plural'. The tagger was originally written by Kristina Toutanova. Since that time, Dan Klein, Christopher Manning, William Morgan, Anna Rafferty, Michel Galley, and John Bauer have improved its speed, performance, usability, and support for other languages.

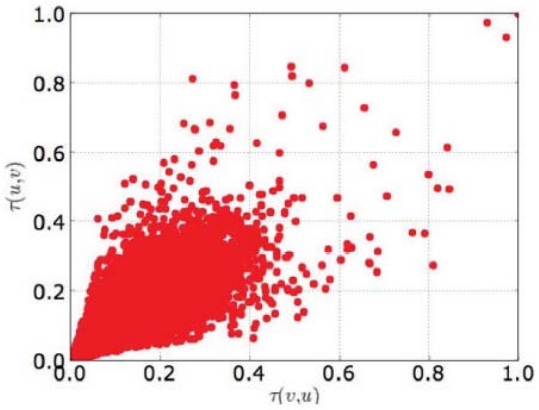
The English taggers use the Penn Treebank tag set. Here are some links to documentation of the Penn Treebank English POS tag. Part-of-speech tagging is harder than just having a list of words and their parts of speech, because some words can represent more than one part of speech at different times, and because some parts of speech are complex or unspoken.

**C. Lexical Elements**

A lexical element refers to a character or groupings of characters that may legally appear in a source file. Predict the sentiment for the lexicon in the format of unigram and bigram. For each lexical element to measure its relatedness to other lexical elements and group related lexical elements to create a sentiment sensitive thesaurus. The lexical elements in the sentiment sensitive thesaurus because when predicting the sentiment label for target reviews. A group related lexical elements to create a sentiment sensitive thesaurus.

The language consists of grammatical zed lexis, and not lexicalized grammar. The entire store of lexical items in a language is called its lexis. A lexical element refers to a character or groupings of characters that may legally appear in a source file. The lexical analysis is the process of converting a sequence of characters into a sequence of tokens.

Function that performs lexical analysis is called a lexical analyzer, lexer, tokenizer, or scanner, though "scanner" is also used for the first stage of a lexer.




Lexers and parsers are most often used for compilers, but can be used for other computer language tools, such as pretty printers or linters. A lexer itself can be divided into two stages: the scanner, which segments the input sequence into groups and categorizes these into token classes; and the evaluator, which converts the raw input characters into a processed value. Tokens are identified based on the specific rules of the lexer. Some methods used to identify tokens include: regular expressions, specific sequences of characters known as a flag, specific separating characters called delimiters, and explicit definition by a dictionary. Special characters, including punctuation characters, are commonly used by lexers to identify tokens because of their natural use in written and programming languages. We compute fðu;wÞ as the point wise mutual information between a lexical element u and a feature w as follows:

$$f(\boldsymbol{u}, w) = \log\left(\frac{\frac{c(u,w)}{N}}{\frac{\sum_{i=1}^{n} c(i,w)}{N} \times \frac{\sum_{j=1}^{m} c(u,j)}{N}}\right). \quad (2)$$

$$\tau(v, u) = \frac{\sum_{w \in \{x | f(v,x) > 0\}} f(\boldsymbol{u}, w)}{\sum_{w \in \{x | f(\boldsymbol{u},x) > 0\}} f(\boldsymbol{u}, w)}. \quad (3)$$

**D. Sentiment Sensitive Element**

As we saw in our example in Section 3, a fundamental problem when applying a sentiment classifier trained on a particular domain to classify reviews on a different domain is that words (hence features) that appear in the reviews in the target domain do not always appear in the trained model. To overcome this feature mismatch problem, we construct a sentiment sensitive thesaurus that captures the relatedness of words as used in different domains. Next, we describe the procedure to construct our sentiment sensitive thesaurus. Given a labeled or an unlabeled review, we first split the review into individual sentences and conduct part-of speech (POS) tagging and lemmatization using the RASP system Lemmatization is the process of normalizing the inflected forms of a word to its lemma. For example, both singular and plural versions of a noun are lemmatized to the same base form. Lemmatization reduces the feature sparseness and has shown to be effective in text classification tasks.

## 6. CONCLUSION AND FUTURE WORK

We proposed a cross-domain sentiment classifier using an automatically extracted sentiment sensitive thesaurus. To overcome the feature mismatch problem in cross-domain sentiment classification, we use labeled data from multiple source domains and unlabeled data from source and target domains to compute the relatedness of features and construct a sentiment sensitive thesaurus. We then use the created thesaurus to expand feature vectors during train and test times for a binary classifier. A relevant subset of the features is selected using L1 regularization. The proposed method significantly outperforms several baselines and reports results that are comparable with previously proposed cross-domain sentiment classification methods on a benchmark data set. Moreover, our comparisons against the SentiWordNet show that the created sentiment-sensitive thesaurus accurately groups words that express similar sentiments. In future, we plan to generalize the proposed method to solve other types of domain adaptation tasks. Identify which source domain features are related to which target domain features. Required a learning framework to incorporate the information regarding the relatedness of source and target domain features. To achieve, propose a fully automatic method to create a thesaurus that is sensitive to the sentiment of words expressed in different domains. The utilize both labeled and unlabeled data available for the source domains and unlabeled data from the target domain using SSN.

## 7. REFERENCES

[1] B. Pang, L. Lee, and S. Vaithyanathan, “Thumbs Up? Sentiment Classification Using Machine Learning Techniques,” Proc. ACL-02 Conf. Empirical Methods in Natural Language Processing (EMNLP ’02), pp. 79-86, 2002

[2] P.D. Turney, “Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews,” Proc. 40th Ann. Meeting on Assoc. for Computational Linguistics (ACL ’02), pp. 417-424, 2002.

[3] B. Pang and L. Lee, "Opinion Mining and Sentiment Analysis," Foundations and Trends in Information Retrieval, vol. 2, nos. 1/2, pp. 1-135, 2008.

[4] Y. Lu, C. Zhai, and N. Sundaresan, "Rated Aspect Summarization of Short Comments," Proc. 18th Int'l Conf. World Wide Web (WWW '09), pp. 131-140, 2009.

[5] T.-K. Fan and C.-H. Chang, "Sentiment-Oriented Contextual Advertising," Knowledge and Information Systems, vol. 23, no. 3, pp. 321-344, 2010.

[6] M. Hu and B. Liu, "Mining and Summarizing Customer Reviews," Proc. 10th ACM SIGKDD Int'l Conf. Knowledge Discovery and Data Mining (KDD '04), pp. 168-177, 2004.

[7] J. Blitzer, M. Dredze, and F. Pereira, "Biographies, Bollywood, Boom-Boxes and Blenders: Domain Adaptation for Sentiment Classification," Proc. 45th Ann. Meeting of the Assoc. Computational Linguistics (ACL '07), pp. 440-447, 2007.

[8] S.J. Pan, X. Ni, J.-T. Sun, Q. Yang, and Z. Chen, "Cross-Domain Sentiment Classification via Spectral Feature Alignment," Proc. 19th Int'l Conf. World Wide Web (WWW '10), 2010.

[9] H. Fang, "A Re-Examination of Query Expansion using Lexical Resources," Proc. Ann. Meeting of the Assoc. Computational Linguistics (ACL '08), pp. 139-147, 2008.

[10] G. Salton and C. Buckley, Introduction to Modern Information Retreival. McGraw-Hill Book Company, 1983.

[11] D. Shen, J. Wu, B. Cao, J.-T. Sun, Q. Yang, Z. Chen, and Y. Li, "Exploiting Term Relationship to Boost Text Classification," Proc. 18th ACM Conf. Information and Knowledge Management (CIKM '09), pp. 1637-1640, 2009.

[12] T. Briscoe, J. Carroll, and R. Watson, "The Second Release of the RASP System," Proc. COLING/ACL Interactive Presentation Sessions Conf., 2006.

[13] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features," Proc. 10th European Conf. Machine Learning (ECML '98), pp. 137-142, 1998.

## AUTHOR BIOGRAPHY

**S. Priya,** ME computer science and Engineering, Infant Jesus College of Engineering and Technology.
priyastalin90@gmail.com.

**Mr. A. Jegadeesh, M.E, (Ph.D),** Assistant Professor, Dept of computer science and Engineering, Infant Jesus College of Engineering and Technology.