Data-Driven Contextual Valence Shifter Quantification for Multi-Theme Sentiment Analysis

Hongkun Yu^{1*}, Jingbo Shang^{1*}, Meichun Hsu², Malú Castellanos², Jiawei Han¹

¹Computer Science Department, University of Illinois at Urbana-Champaign, IL, USA
{hyu50, shang7, hanj}@illinois.edu

²HPE Vertica, CA, USA
{meichun.hsu, malu.castellanos}@hpe.com

ABSTRACT

Users often write reviews on different themes involving linguistic structures with complex sentiments. The sentiment polarity of a word can be different across themes. Moreover, contextual valence shifters may change sentiment polarity depending on the contexts that they appear in. Both challenges cannot be modeled effectively and explicitly in traditional sentiment analysis. Studying both phenomena requires multi-theme sentiment analysis at the word level, which is very interesting but significantly more challenging than overall polarity classification.

To simultaneously resolve the *multi-theme* and *sentiment shifting* problems, we propose a data-driven framework to enable both capabilities: (1) polarity predictions of the same word in reviews of different themes, and (2) discovery and quantification of contextual valence shifters. The framework formulates multi-theme sentiment by factorizing the review sentiments with theme/word embeddings and then derives the shifter effect learning problem as a logistic regression. The improvement of sentiment polarity classification accuracy demonstrates not only the importance of *multi-theme* and *sentiment shifting*, but also effectiveness of our framework. Human evaluations and case studies further show the success of multi-theme word sentiment predictions and automatic effect quantification of contextual valence shifters.

Keywords

Sentiment Analysis; Multi-Theme; Sentiment Shifting

1. INTRODUCTION

With the proliferation of social media and interactions between businesses and customers, the daily growth of online reviews has become explosive. Due to the infeasibility of timely human analysis, automatic polarity classification for reviews, including product/movie reviews, comments

*Equal Contribution.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee. Request permissions from permissions@acm.org.

CIKM'16, October 24-28, 2016, Indianapolis, IN, USA

© 2016 ACM. ISBN 978-1-4503-4073-1/16/10...\$15.00

DOI: <http://dx.doi.org/10.1145/2983323.2983793>

and microblogs, has become an important research topic attracting much attention. The polarity classification problem [24] has been well studied by various machine learning techniques and recently some state-of-the-art and complex neural network-based models achieved superior accuracy over review datasets [30, 8]. However, in many real world scenarios, businesses are interested in not only the overall polarity of a review, but also how sentiment words contribute to the overall polarity in a certain context and theme, which cannot be provided by those complex and less interpretable neural network-based models.

There are two major challenges when delving into the fine-grained sentiment analysis: *multi-theme* and *sentiment shifting*, as shown in Figure 1. To resolve both challenges, we are motivated to study multi-theme sentiment analysis at the word level. Owing to complexity of reviews, inconsistency of word sentiment in different themes and sentiment shifting, the task is very interesting but significantly more challenging than overall polarity classification. As a result, the bag-of-words model still offers the most interpretable feature space that is suitable for detailed sentiment analysis beyond document-level classification.

Figure 1: Examples of multi-theme and sentiment shifting challenges.

The review documents often consist of multiple themes (*multi-theme*), where a theme is an abstract concept, such as an aspect (e.g., the service or food quality of a restaurant) or a category (e.g., thriller movies vs. cartoon movies), as described in [20]. A word may convey different degrees of polarity when being used in different themes. For example, as shown in Figure 1, “long” is a positive word when describing the life of a battery, but becomes negative if talking about a waiting queue in a restaurant. Similarly, “thrilling” should be a positive word in the reviews of thriller movies, but it usually reveals a negative sentiment in the reviews of other categories of movies. The inconsistency of word sentiment in different themes becomes the most challenging obstacle for traditional sentiment analysis techniques when applying to review documents containing multiple themes.

Recalling sentiment analysis techniques, plain discrimina-

tive machine learning models using the bag-of-words representations learn the sentiment of sentiment words, but each word has a fixed sentiment polarity even in different themes [24]. Lexicon-based approaches rely on external knowledge and the lexicon quality may vary depending on the reviews in different domains [36, 17]. Although overall sentiment polarity or ratings on specific topical themes has been studied [32, 33], direct analysis on *multi-theme* challenge for individual words or phrases still lacks research.

Regarding another critical issue, contextual valence shifters [25] with *sentiment shifting* effects such as negation, intensification, and diminishment, are the common factors that affect other words inside a context when expressing sentiments. The presences of the *sentiment shifting* may interfere the learning of word polarity scores. For example, as shown in Figure 1, if using the bag-of-words model, without noticing the effects of negation words, e.g., “*not*” in sentences like “*I am not very happy*”, the polarity score of “*happy*” will be skewed and even reversed if multiple occurrences of “*happy*” are modified by negation words.

Natural Language Process (NLP) community has noticed the importance of shifters to sentiment analysis and attempted various rules to incorporate shifter-related features into the feature representations of data [9, 37] with insignificant improvements in sentiment classification tasks. However, rule-based methods have limited ability to discover new sentiment shifting phenomena, quantify the effects of shifters, or take the *multi-theme* challenge into consideration. [5] adopts classifiers to determine sentiment shifting for polarity terms depending on the entire sentence, but it relies on domain knowledge of sentiment dictionary. Other studies [10, 2] fail to quantify shifter effects despite shifters extraction methods using syntactic structures.

We propose to study the problem of sentiment analysis for review documents focusing on *multi-theme* and *sentiment shifting* challenges without external knowledge (e.g. a sentiment lexicon), because (1) human efforts on fine-grained labeling are tremendously expensive; and (2) without noticing effects of shifters, it is almost impossible to precisely and consistently identify the sentiment polarity of a single word in different themes, and without precise polarity estimations, we can hardly learn effects of shifters. Our goal is that, based on (binary) sentiment polarity labels of review documents, we should identify how words impose interpretable effects if different reviews have various themes and quantify effects of shifters. To the best of our knowledge, this is the first effort to address both *multi-theme* and *sentiment shifting* challenges simultaneously using data-driven approaches.

In this paper, we propose a general iterative multi-theme sentiment analysis framework, **MTSA**, to address both challenges at the same time. To be more specific, MTSA formulates multi-theme sentiment by factorizing the review sentiments and learning embeddings (i.e., vector representations) for both themes and words. Leveraging theme embedding and word embedding, MTSA is able to examine whether a word has consistent sentiment over different themes and then MTSA constructs confident training contexts for shifter candidates and derives the shifter effect learning algorithm by modeling shifted sentiments in a logistic regression model. By iteratively feeding the output of each component into the other, the embeddings will be further improved by the rectified document feature representations, and vice versa.

The main contributions of this paper are as follows.

- We propose a series of intuitive assumptions to rigorously formulate the *multi-theme* and *sentiment shifting* challenges, and then simultaneously resolve both challenges through a data-driven approach.
- We propose a data-driven sentiment analysis model, MTSA, to learn theme and word embeddings capturing different sentiment polarities of the same word in different themes, and automatically discover and quantify contextual valence shifters based on training data. The joint MTSA model enables mutual enhancement between two components.
- We carefully verify the proposed assumptions and effectiveness of MTSA by extensive experiments based on real-world datasets.

2. PROBLEM DEFINITION

This paper deals with multi-theme sentiment analysis and sentiment shifting in a large collection of reviews given the sentiment label for each review. We first formulate the problem as follows and explain the concepts one by one.

DEFINITION 1. *Given a review corpus $\mathcal{D}$, as well as the sentiment label (polarity or score) y_r and the theme descriptor θ_r for each review $r \in \mathcal{D}$, the task is to: (1) model different sentiment polarities of the same word in different themes; (2) discover the shifters and quantify their effects in $\mathcal{D}$ and rectify the word-level descriptors $\mathbf{W}$ for documents; and (3) derive a unified sentiment analysis framework, taking advantage of multi-theme modeling and shifters.*

We define the particular themes in this study as:

DEFINITION 2. *For any review collection, **themes** are a fixed number of latent conceptual categories where word sentiment may vary across themes.*

The themes in each review r are described as a vector representation θ_r , where θ_{ri} is the weight of theme i in the review r . The representation is capable to describe not only different aspects (e.g., service and environment for restaurants) but also different categories of review target (e.g., horror movie and romantic movie).

In this work, the themes are assumed to be either extracted beforehand via any rule-based or learning-based theme discovery methods or given through extra meta knowledge. For example, $\Theta \in \mathbb{R}^{K \times |\mathcal{D}|}$ denotes the column-wise concatenated matrix of the theme descriptors of all reviews, where K is the number of themes.

Contextual valence shifters (“*shifters*” for short, in this paper), a family of linguistic patterns, are very popular—more than 90% review documents contain at least one shifter in the review corpora we examined. As described in the linguistic study [9], shifters usually have three different types: negations that reverse the semantic polarity of a particular term, intensifiers that increase the degree to which a term is positive or negative, and diminishers that decrease the degree.

DEFINITION 3. *A **shifter** is a word/phrase w that has an effect (negation, intensification, diminishment) on the sentiment polarities of its context words/phrases C_w .*

Our framework is flexible with different sentiment context identification methods. Although semantic parser may identify useful long-range dependencies, few existing systems

can directly achieve our goals and categorizing dependencies by hand-crafted rules makes it not data-driven. With a data-driven fashion, we choose the sliding window due to its simplicity and robustness for model performance, which is testified in experiments. That is, for the i -th word w_i in a sentence, its context C_{w_i} is the word sequence from $(i - \Delta_l)$ -th position to $(i + \Delta_r)$ -th position, where Δ_l and Δ_r are the parameters of the sliding window.

The bag-of-words model is one of the most friendly models to incorporate shifters, because once we figure out their effects in the contexts they belong to, we can rectify the word-level descriptors $\mathbf{W}$ and refine the classification model based on the rectified descriptors, as presented in § 3.4.

MTSA can be divided into two iterative phases: (1) the theme and word embedding learning phase, which discovers the *multi-theme* phenomenon for sentiment words and extracts common sentiment words being invariant across themes, given theme descriptors Θ and word-level descriptors $\mathbf{W}$; and (2) the shifter effects learning phase, which learns the valid quantified shifters from candidates and applies them to correct word-level descriptors $\mathbf{W}$, as outlined below:

- *Theme and Word Embedding Learning*

1. Learn latent vector representations for themes and words as their embeddings by optimizing an objective function with respect to sentiment polarity label of each training review, given observed review content and extracted theme descriptors.
2. Through ranking sentiment words by estimated sentiment polarities under each theme, extract the common (i.e., theme-invariant) sentiment words which are ranked as top positive or negative in most themes.

- *Shifter Effects Learning*

1. Learn effects for the shifter candidates (including phrase candidates) generated from training reviews by optimizing an objective function with respect to sentiment polarity label of each training review, given extracted common sentiment words as well as the learned theme and word embeddings.
2. Apply learned shifter effects to the contexts they have affected and update word-level descriptors $\mathbf{W}$ accordingly.

3. METHODOLOGY

3.1 Descriptors

Theme Descriptors. In this paper, we adopt a probabilistic distribution over the themes as a theme descriptor of each review and assume that the content of a review is generated from a multinomial distribution. More specifically, $\theta_{ri} = Pr(t = i|r)$ describes how much different themes are emphasized in the review r . Nonetheless, in future extension, it is interesting and natural to incorporate knowledge from external resources to form theme descriptors. For example, a taxonomy hierarchy for reviews can be encoded as binary indicators.

Word-Level Descriptors. Similarly, word-level vector representation (e.g., bag-of-words) is adopted as the word-level descriptor $\mathbf{W}_r \in \mathbb{R}^{|\Sigma|}$ of each review r , where $W_{rj} \in \mathbb{R}$ indicates the weight of word j in review r . Note that the word-level descriptors $\mathbf{W}$ are not restricted to bag-of-words

and can be replaced by any other descriptors or combined with others.

In our iterative framework, in order to be consistent with shifter effect learning step, we adopt a general tf-idf weighting scheme for $\mathbf{W}$. To guarantee W_{rj} to be a linear combination of contributions of tokens which may be negated, diminished or intensified by shifters, instead of term frequency functions for word j , we choose to consider each occurrence of word j . More specifically, for a single occurrence of word j in review r , the increment of W_{rj} is $\delta(r, j) = \Delta tf(r) \times idf(j)$, where $\Delta tf(r)$ denotes the normalized increment of term frequency in review r and $idf(j)$ denotes inverse document frequency of word j in the corpus $\mathcal{D}$.

3.2 Multi-Theme Review Sentiment Modeling

Being aware of *multi-theme*, the MTSA model follows the assumption below.

ASSUMPTION 1. (Multi-theme Assumption). *The sentiment polarity of a word in different themes may be different.*

Inspired by the *Multi-theme Assumption*, we desire to learn expressive sentiment embeddings for each review theme i and word j , by introducing vector representations $\mathbf{p}_i, \mathbf{q}_j \in \mathbb{R}^d$ and defining the sentiment polarity for word j in theme i as:

$$s_{ij} = \mathbf{p}_i^T \mathbf{q}_j \quad (1)$$

For a review r , we model its sentiment polarity, s_r , by combining the affects of different words in different themes as:

$$s_r = \sum_{i=1}^K \theta_{ri} \sum_{j=1}^{|\Sigma|} W_{rj} s_{ij} = \sum_{i=1}^K \sum_{j=1}^{|\Sigma|} (\theta_{ri} \mathbf{p}_i)^T (W_{rj} \mathbf{q}_j) \quad (2)$$

Let $\mathbf{P} \in \mathbb{R}^{d \times K}$ denote the matrix of theme embeddings for all K themes, and $\mathbf{Q} \in \mathbb{R}^{d \times |\Sigma|}$ denote the matrix of word embeddings for the whole vocabulary Σ . By using matrix operations, the prediction is equivalent to the following equation.

$$s_r = (\mathbf{P} \Theta_r)^T (\mathbf{Q} \mathbf{W}_r) \quad (3)$$

Since most polarity labels are binary, by taking the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}}$ for sentiment s_r , we have the prediction $\hat{y}_r = \sigma(s_r)$ for review r with probability interpretation. Thereby, the log likelihood of the whole review corpus $\mathcal{D}$ formulates the objective of the optimization as:

$$\mathcal{L} = \sum_{r=1}^{|\mathcal{D}|} y_r \ln \hat{y}_r + (1 - y_r) \ln(1 - \hat{y}_r). \quad (4)$$

Besides probabilistic interpretation, other loss functions such as square loss and hinge loss are also adaptable in our model.

To avoid overfitting on observed data, we adopt the elastic net [40] as the regularization to guarantee sparsity on sentiment embeddings. Therefore, we can formulate our objective as a combination of sentiment objective as Eq. (4) and regularization on embeddings:

$$\mathcal{O} = \mathcal{L} + \Omega(\mathbf{P}) + \Omega(\mathbf{Q}) \quad (5)$$

where $\Omega(\mathbf{P}) = \alpha_1 \|\mathbf{P}\|_1 + \frac{\alpha_2}{2} \|\mathbf{P}\|_2^2$, and α_1 and α_2 are two parameters controlling the weights of regularization terms.

Due to the limit of space, we do not derive the optimization algorithm that learns $\mathbf{P}$ and $\mathbf{Q}$ through minimizing the objective $\mathcal{O}$. Any mature optimization technique, such as coordinate descent or stochastic gradient descent, can be adopted.

3.3 Sentiment Shifter Modeling

In most vector space models, each review document is represented as a bag-of-words feature vector describing the (tf-idf weighted) counts of words. However, in almost all human languages, a simple n -gram model cannot correctly represent the underlying meaning or sentiment of the document. As an example, to capture the sentiment of “*The beef does not really look as spicy as ...*”, we need to be aware that “*not*” serves as a negation shifter. The MTSA model captures these shifters by following the *Theme-invariant Assumption*.

ASSUMPTION 2. (Theme-invariant Assumption). *The effect of a shifter is consistent in different themes.*

As a result, to discover shifters, we leverage the theme-invariant sentiment words, a subset of sentiment words, which have consistent polarity in almost all themes in our model.

In this paper, learning effects of shifters is to assign the most appropriate quantified effects that achieve an accurate polarity classification in the given corpus $\mathcal{D}$, instead of only categorizing the shifters into negation/intensifier/diminisher. Therefore, we formally define the real-value effect of a shifter and propose the *Independent-effect Assumption* to incorporate the shifter effect into our MTSA model, as follows.

DEFINITION 4. *For a shifter w , its effect is modeled as a real value $f_w \in \mathbb{R}$.*

ASSUMPTION 3. (Independent-effect Assumption). *If the word at i -th position in the review r is within the contexts of m shifters $w_1, \dots, w_m$, its shifted polarity, $s'_i = s_i \cdot \prod_{k=1}^m f_{w_k}$ where its original polarity is $s_i = \sum_{j=1}^K \theta_{rj} s_{ij}$.*

The rationales behind *Independent-effect Assumption* are: (1) we need to quantitatively define the shifter effects; (2) many such cases exist in real data; (3) the problem is simplified and thus solvable. Admittedly, studying complicated composition of shifters is an interesting future work. We are aware of possible complicated composition and thus propose to mine phrase shifters in § 3.3.1 to alleviate the issue and the strategy works decently in practice.

3.3.1 Shifter Candidates Generation

As we assumed there exist a set of shifters Σ_s for every language, an automatic method generating the potential candidates is a necessity rather than merely using human efforts. A set of negation keywords in English are defined in [6]. However, beyond such a thorough linguistic study of negation keywords, there might be more phrases that can express the shifting effect to different extents. Besides prepositions, in real cases, verbs, adverbs, and other types of words may also serve as shifters. Due to the difficulty of directly discovering such candidate sets from data, we propose a compromised bootstrapping solution by leveraging the contextual similarity between words defined by word embedding [21].

Our proposed method starts from a very small and confident set of shifter candidates as seeds, like negation words “*never*” and “*not*”, as well as intensifier/diminisher words “*very*” and “*extremely*”, and then enlarges the candidate set by retrieving the nearest words in terms of contextual similarity (e.g., cosine similarity defined by word embeddings) to the current candidates. Empirically, we find it is much better to discover negation and intensifier/diminisher as two

separate sets with some overlapping. More specifically, we directly leverage an open-source reliable model, word2vec [21] trained from Google News dataset to calculate contextual similarity, since such unigram shifters objectively exist inside language. Alternatively, the word embeddings may also be trained on the review corpus but data could be insufficient. The contextual similarity is usually defined as cosine similarity: for words i and j in vocabulary, we have their vector representations as $\mathbf{v}_i$ and $\mathbf{v}_j$ in word2vec model, and the contextual similarity, $Sim(i, j) = \frac{\mathbf{v}_i \cdot \mathbf{v}_j}{\|\mathbf{v}_i\|_2 \|\mathbf{v}_j\|_2}$. After running bootstrapping process several times, we finally obtain two candidate sets Σ_n (123 potential negations) and Σ_r (153 potential intensifiers/diminishers) with manual cleaning for candidates in obvious wrong parts of speech. As our algorithm will finally learn the effects of shifters from training data, we are able to resolve the noise and uncertainty in the initial sets by removing and adjusting these shifters based on their learned effects.

Furthermore, the unigram shifters cannot perfectly represent the nature of the language like English, where many multi-word phrases have the effect of negation, intensification or diminishment. To address this problem, i.e., multi-word phrases such as “*not very*” and “*too ... to ...*”, we propose to utilize frequent pattern mining [4] to discover phrase shifters Σ_p . Because allowing non-consecutive phrases will exponentially increase the complexity of the problem, our method only considers the phrases composed by consecutive unigram shifters as potential phrase shifters. Σ_p accepts the frequent phrases whose frequency meet a threshold proportional to the corpus size and that satisfy simple rules.

Combining candidate bootstrapping and phrase shifters mining together, three sets of shifter candidates are generated as $\Sigma_n, \Sigma_r, \Sigma_p$ and $\Sigma_s = \Sigma_n \cup \Sigma_r \cup \Sigma_p$.

Figure 2: Sentiment shifter learning process

3.3.2 Shifter Effect Learning

In order to obtain precise shifters, we learn the effects of shifters only based on (1) common sentiment words, and (2) single-modification contexts. The reasons are as follows.

Since shifter learning requires accurate learned polarities of sentiment words, considering only common theme-invariant sentiment words, when constructing shifter contexts, is a safer choice to learn reliable effects of shifters, according to *Theme-invariant Assumption*. Thanks to multi-theme sentiment modeling, the common sentiment words, such as “*amazing*” and “*awful*”, are able to be automatically extracted and such set of common sentiment words is denoted as Σ_c . More specifically, given theme embeddings $\mathbf{P}$ and word embeddings $\mathbf{Q}$, the common sentiment words in Σ_c should be ranked in the top- N positives or top- N negatives based on theme polarities defined in Eq. (1) in at least M over K themes. For example, in terms of the positive common sentiment words, we may choose top-150 positive words from all themes and select the words that present in at least 12 over 20 themes.

Caused by the *Independent-effect Assumption*, when a

Algorithm 1: Shifter Effect Learning

Input: review documents $\mathcal{D}$, a vector of quantified effects $\mathbf{f}^{t-1}$, review feature matrix $\mathbf{W}$, parameters matrices $\Theta, \mathbf{P}, \mathbf{Q}$, common sentiment words Σ_c
for each review r in $\mathcal{D}$ do
 for each shifter w in Σ_s do
 Identify sentiment contexts C_w of w in r
 Identify the single modified words in C_w and the words must be in Σ_c
 Construct shifter features in $\mathbf{x}_r$ by Eq. (7)
Solve Eq. (9) for $\mathbf{f}^*$
Return: a vector of quantified effects $\mathbf{f}^*$, which serves as $\mathbf{f}^t$ for next iteration

sentiment word i lies in the context of m ($m \geq 2$) shifters $w_1, w_2, \dots, w_m$, e.g., “*not really good*”, the shifted polarity s'_i of this sentiment word relies on the product of unknown effects $\prod_{k=1}^m f_{w_k}$. Such complex dependency makes the learning of these quantified effects of shifters unstable and intractable, thus substantially increasing the difficulty of the learning process. Therefore, despite the sacrifice of some training data, we simplify the learning process by only considering single-modification contexts where every sentiment word is just modified by a single shifter.

As an iterative framework, MTSA updates shifter effects after each iteration of multi-theme modeling. Next, assuming at the t -th iteration, taking the previous estimated effect of a shifter w as f_w^{t-1} and the true effect of a shifter w as an unknown variable f_w^* , we will derive the rectified sentiment. For a particular occurrence of sentiment word j in review r , assume word j is modified by a shifter w . In general, the contribution of this modified occurrence of word j to word-level descriptor W_{rj} should be $\delta(r, j) \times f_w^*$, where $\delta(r, j) = \Delta t f(r) \times idf(j)$, but currently the contribution is miscalculated as $\delta(r, j) \times f_w^{t-1}$. Under tf-idf weighting scheme, $\Delta t f(r)$ denotes the normalized increment of term frequency in review r and $idf(j)$ denotes inverse document frequency of word j in the corpus $\mathcal{D}$. Therefore, considering a single modified occurrence of word j , whose sentiment is $(\theta_r \mathbf{P})^T \mathbf{q}_j$, the rectified sentiment of review r should be :

$$s'_r = s_r + (f_w^* - f_w^{t-1}) \times \delta(r, j) (\theta_r \mathbf{P})^T \mathbf{q}_j \quad (6)$$

Then, we take the rectified sentiment s'_r instead of s_r to predict review level sentiment. After rewriting Eq. (6) for all modified words in reviews, it naturally formulates a logistic regression problem. The feature vector of review r as $\mathbf{x}_r \in \mathbb{R}^{|\Sigma_s|}$ is:

$$\mathbf{x}_{rw} = \sum_{j \in C_{rw}} \delta(r, j) (\theta_r \mathbf{P})^T \mathbf{q}_j \quad (7)$$

where C_{rw} is the context of shift w in review r .

The logistic regression problem is formulated as follows.

$$\begin{aligned} \hat{y}_r &= \sigma(\mathbf{f} \cdot \mathbf{x}_r + \bar{s}_r) \\ \bar{s}_r &= s_r - \sum_{w \in \Sigma_s} f_w^{t-1} \sum_{j \in C_w} \delta(r, j) (\theta_r \mathbf{P})^T \mathbf{q}_j \end{aligned} \quad (8)$$

The l_2 regularization $\Omega(\mathbf{f}) = \alpha \sum_{w \in \Sigma_s} (f_w - 1)^2$ is added to avoid fitting skewed data and obtain reasonable quantified effects around 1.0 which indicates that this shifter has no effect on sentiment words. Thereby, we have the objective

function to minimize and solve for $\mathbf{f}$:

$$\mathcal{O}_{\mathbf{f}} = \sum_{r=1}^{|\mathcal{D}|} y_r \ln \hat{y}_r + (1 - y_r) \ln(1 - \hat{y}_r) + \alpha \sum_{w \in \Sigma_s} (f_w - 1)^2 \quad (9)$$

We utilize the optimal solution of the above logistic regression problem as $\mathbf{f}^t$ for further iteration.

Thus, the work flow of one iteration for shifter effect learning in the whole MTSA framework is described in Algorithm 1 and an illustrative example is shown as Figure 2. This model can be viewed as: figuring out how to put the correct sentiment effects of the words inside contexts of shifters back by learning appropriate shifter effects. Through the steps described above, the effects of shifters are learned as real-value vectors: $\mathbf{f}_n$ for negation Σ_n , $\mathbf{f}_r$ for intensifier/diminisher Σ_r , and $\mathbf{f}_p$ for shifter phrases Σ_p , as long as they are learned to be valid shifters from the data.

3.4 Rectification and Iterative Refinement

After learning the quantified effects of shifters by Algorithm 1, we obtain three sets of useful shifters and their effects. The word-level descriptors $\mathbf{W}$ are ready to be rectified by plugging in the effects of these learned shifters. More specifically, instead of summing over the word occurrences (equivalent to $\mathbf{f} = \mathbf{1}$) for the computation of $\mathbf{W}$, we now identify the context for each shifter w and apply its effect f_w on the words within the context.

In this step, we are no longer restricted to common sentiment words and single-modification contexts, since (1) the word-level descriptors $\mathbf{W}$ contains all words instead of common sentiment words only; and (2) the overall effect of multiple shifters modifying the same sentiment word can be accumulated as the product of shifter effects based on *Independent-effect Assumption*.

As tf-idf weighting scheme introduced in § 3.1, for review r and word j , assuming O_{rj} contains all occurrences of word j in review r , we examine if word j presents in any context of shifters and have:

$$W_{rj} = \delta(r, j) \times \sum_{k \in O_{rj}} \prod_{w \in \Sigma_s \wedge w \text{ modifies } k} f_w \quad (10)$$

Note that, the plain tf-idf feature is a special case when $\mathbf{f} = \mathbf{1}$.

Benefiting from the learning of the multi-theme sentiment model and quantified shifter effects, our sentiment polarity classification can be iteratively refined. In our framework, the learned theme and word embeddings, $\mathbf{P}$ and $\mathbf{Q}$, together define the multi-theme sentiment polarity classifiers. Although the theme descriptors Θ are pre-computed and fixed for reviews, the bag-of-words features $\mathbf{W}$ are adjusted once the effects of shifters are updated. In the procedure of rectifying $\mathbf{W}$ and training more accurate embeddings $\mathbf{P}$ and $\mathbf{Q}$, classification performance is improved via mutual enhancements, although there is no guarantee on the convergence due to the possible noises in the data.

3.5 Computational Complexity Analysis

Suppose the number of tokens in the review corpus $\mathcal{D}$ is $N_{\mathcal{D}}$, the size of the vocabulary is $|\Sigma|$, and the number of reviews is $|\mathcal{D}|$. The feature extraction in shifter effect learning in Algorithm 1 is $O(N_{\mathcal{D}})$ as the size of the shifter set $|\Sigma_s|$ can

Table 1: Datasets Description

	RT	Yelp	IMDB
total instances	10,662	30,471	50,000
test instances	N/A	8,341	25,000
avg. review length	22	137	241
theme descriptor dim	5	20	20
theme extraction	BTM	LDA	LDA
evaluation	10-fold CV	test set	test set

be viewed as a small constant. The model learning for multi-theme sentiment model takes $O(d(K + |\Sigma| + |\mathcal{D}|))$, where d is the dimensionality of embeddings, for each coordinate descent iteration, if we employ the speedup trick in [27]. The step of shifter effect learning as a logistic regression in Algorithm 1 is fairly efficient comparing to multi-theme sentiment learning step.

4. EXPERIMENTS

We design experiments to demonstrate benefits from modeling both *multi-theme* and *sentiment shifting* by showing the improvement of classification accuracies on three real world datasets comparing to MTSA variants and several baselines. Besides, human evaluation, case study and visualization are introduced to verify the proposed assumptions and show effectiveness of MTSA.

4.1 Datasets and Settings

Our experiments use three real-world datasets as summarized in Table 1.

- **Rotten Tomatoes (RT)** snippets dataset¹, introduced in [23], contains even number of positive and negative short reviews. 10-fold cross validation is reported in literature.
- **Yelp** dataset contains 5-scale rated reviews with timestamps selected from the Yelp Challenge dataset², including businesses in the categories of restaurants, shopping centers, automotive, gym and drinks & bars. A negative review has a score no more than 2, while a positive review has a score at least 4. The dataset is partitioned into training and testing by a split point of time: 2014-08-01. The test set contains 3,779 positive examples and 4,562 negative examples.
- **IMDB** dataset³ is introduced in [18] as a benchmark for sentiment analysis, where the positive and negative reviews are balanced.

Theme Descriptor. In our model MTSA, for Yelp and IMDB with long reviews, the LDA implementation in MALLET [19] is adopted to obtain theme descriptors with a fixed number of themes $K = 20$ for both training and testing sets. Since reviews in RT are too short for LDA to estimate the posterior topic distributions, we adopt a bitern topic model (BTM) for short text [39] and fixed the number of themes $K = 5$. Infrequent words and stopwords are removed before descriptor extraction.

Word-Level Descriptor. The non-literal characters are removed in Yelp and RT, while the vocabulary used in the IMDB dataset is synchronized with the suggestion in [18], where characters composing emoji like “smile” face are preserved. As we proposed a general framework, different idf scores can be applied, because of various ways to achieve

tf-idf weighting in literature. Besides the log-scaled inverse fraction, Navie Bayes feature (NB) was proposed in [35], which performs very well for sentiment classification tasks. We will show that the NB feature is actually an idf weight function considering training labels. Define the count vectors as $\mathbf{p} = \alpha + \sum_{i:y^i=1} \mathbf{W}_r$ and $\mathbf{q} = \alpha + \sum_{r:y^r=0} \mathbf{W}_r$ where α serves as smoothing. [35] advises to use binary feature for $\mathbf{W}_r$, and defines the log ratio $\mathbf{r}$ for words and NB feature as:

$$\mathbf{r} = \log\left(\frac{\mathbf{p}/\|\mathbf{p}\|_1}{\mathbf{q}/\|\mathbf{q}\|_1}\right), \quad \hat{\mathbf{W}}_r = \mathbf{r} \odot \mathbf{W}_r \quad (11)$$

where $\odot$ indicates element-wise product. Because the log ratio $\mathbf{r}$ is applied on all features, if we take absolute value of each element in $\mathbf{r}$ to obtain $abs(\mathbf{r})$, it serves as a non-negative idf function which measures how discriminative a feature is. Since $\mathbf{r}$ is globally multiplied to all instances, taking absolute value will not change predictions. Being consistent with the reported baselines, for all bigram baselines, binary presents are chosen to be the best performed features.

Parameter Tuning and Evaluation. The parameter tuning scheme for all compared baseline methods and for our model is as follows. A validation set is held-out in the training data and the final classification accuracy is evaluated on the test data. For our approach, the multi-theme model is tuned at the first step. After fixing the parameters in multi-theme model, we further tune the parameters related to shifter learning. The dimensionality of embeddings is empirically set as $d = 64$ for both Yelp and IMDB, while $d = 32$ is chosen for RT because the average number of tokens in a review is much less. Based on our experiments and for reducing the risk of overfitting, we choose to first perform four rounds of cold-start learning, which optimizes the model starting from random initialization, and then apply one more round of warm-start learning, which takes the previous model as initialization. This setting usually yields a reasonably good accuracy. Our implementation utilizes the feature-based matrix factorization toolkit PL2M [27] for the multi-theme model and L-BFGS [22] for optimization in the shifter learning process.

Shifter Context. As defined in Definition 3, there are various ways in practice to extract contexts of a shifter. Although some semantic parsing might be more precise to determine semantic scope, the context is extracted in our experiment by checking a fixed sized sliding window (e.g., size 3, (0, 3]). In practice, simple sliding window may bring much noise when applying the learned shifters to contexts. To alleviate the incorrect attachment, the only heuristic rule we adopt in the experiment is that we place adjectives with higher priority, and only if no adjective exists in the window, we take the immediate successive term as the back-off choice.

4.2 Review Sentiment Classification

MTSA can be applied for review sentiment polarity classification. We compare our algorithms with other baselines on the binary (positive/negative) classification task. It is worth mentioning that our model is very flexible to be adapted for review score regression tasks.

The baselines chosen to compare with MTSA are popular discriminative classifiers and NBSVM [35] in bag-of-words feature space. As our model incorporated logistic loss function, we also present the accuracies of logistic regression model in experiment. To have fair comparisons, NB features are also compared in various models. The results of

¹www.cs.cornell.edu/people/pabo/movie-review-data/

²www.yelp.com/dataset_challenge

³ai.stanford.edu/~amaas/data/sentiment/ Additional 50,000 unlabeled reviews are not used in our experiment.

Table 2: Comparison of classification accuracy (%).

	Algo. & Features	RT	Yelp	IMDB
unigram	Linear SVM	76.2 *	91.87	87.8 †
	Logistic	76.9	91.90	88.19
	NBSVM	78.1 *	92.13	88.29 *
	MTSA	78.0	92.20	88.57
	MTSA(fixed negation)	78.3	92.20	88.48
	MTSA(NB)	78.3	92.52	88.81
	MTSA(shifter)	78.4	92.78	88.82
	MTSA(NB + shifter)	78.8	93.08	88.97
bigram	Linear SVM	77.7 *	91.93	89.16 *
	Logistic	78.1	92.99	89.18
	NBSVM	79.4 *	93.99	91.22 *
	MTSA(NB + shifter) [‡]	81.3	94.07	90.44

† reported in [18]; binary features with cosine normalization (bnc).

* reported in [35]; NBSVM is an ensemble method.

‡ The top bigram features from Logistic bigram are appended to feature vectors.

Figure 3: Accuracies on subsets of testing data. The instances are ranked by the ratio (number of shifters / number of tokens), from high to low.

our experiment and reported benchmarks using bag-of-words representations in literature are shown in Table 2.

Multi-Theme. The performance improvements of MTSA and MTSA(NB) over Linear SVM, Logistic, and NBSVM imply that modeling *multi-theme* is important.

Sentiment Shifting. When adding iterations of shifter learning, the rectified features take advantage of quantified shifters and help MTSA (with “shifter”) achieve further improvement. Thus, unigram MTSA has better accuracy on all three datasets than unigram baselines. Using negation phrases described in [6] as fixed negation effects (i.e. -1) in MTSA(fixed negation) may not improve accuracy. The overall accuracy may not properly illustrate the improvements, because the classification result might be correct without noticing shifters, if there is only few shifter in a review, especially in a long review. Therefore, we sort the testing reviews decreasingly by the ratio of the number of shifters and the number of tokens. In such order, the highly ranked reviews have large portion of tokens that are shifters and the sentiment of these reviews are more likely to be twisted by shifters. We calculate the accuracy@N, the classification accuracy of top N instances. As shown in Figure 3, in Yelp dataset, we can clearly see that when the ratio getting bigger, shifters exist in the review with a larger portion and the gain of modeling shifter effect is bigger as the gap between MTSA and NBSVM gets bigger, which demonstrates MTSA learns accurate shifter effects.

Bigrams. Discussed in several studies, n-gram features

Figure 4: Accuracy improves through iterative refinement of MTSA(unigram)

Table 3: Comparison with newest deep neural models.

Algo.	RT	Algo.	IMDB
CNN [11]	81.5	Paragraph Vec [12]	92.58
SA-LSTM [3] [†]	80.7	SA-LSTM [3]	92.76
MTSA	81.3	MTSA	90.44

† reported without external corpus. 10-fold cross validation is not conducted.

may have better classification performance depending on tasks [35]. Although our MTSA is derived from unigram bag-of-words, we experiment with appending top 3% discriminative bigrams into our learning framework. The performance is boosted by the ad-hoc treatment and the benefit of shifters is more significant on Yelp and RT datasets with shorter reviews. Overall, our MTSA model supersedes compared non-ensemble baselines and achieve the best results on RT and Yelp datasets.

Mutual Enhancement. In Figure 4, one can observe the mutual enhancement between multi-theme modeling and shifter learning through the improvements of classification with respect to the number of iterations, although it may suffer from overfitting when more iterations are performed.

Neural Network-based Methods. Recent years, deep neural network-based methods achieve successful results for various classification task. The family of neural network methods for NLP generally benefit from pre-trained word embeddings as the input and train extensive deep model for the data. However, we did not compete with this family of models on classification performance, because they require time-consuming training even on a small amount of data and lack ability to explain at the word-level while we are studying using quantified shifters to make explainable multi-theme sentiment analysis. According to the literature [12], the studies of neural networks achieved error rates around 7.5% for IMDB dataset as the comparison in Table ?? . We can observe that MTSA has comparable performance on short reviews, RT dataset, comparing to state-of-the-art deep neural models and even supersedes some previous deep neural models such as MV-RNN in [29]. Nevertheless, in terms of long reviews, deep learning seems to perform slightly better than bag-of-words models.

All these models [12, 3] are reported to benefit from unlabeled data to improve performance with unsupervised component. For the long review documents and large training corpora, deep neural models may be able to learn good representations but require very careful tuning on hyper-parameters and massive parameters to learn, e.g. each training review has a embedding vector to learn [12].

4.3 Shifter Discovery and Quantification

To verify the *Theme-Invariant Assumption* and *Independent-effect Assumption*, we ask human experts to examine whether the shifters are correctly discovered and whether their effects

Table 4: Precision of sentiment shifting. *avg*: average precision of judges; *voted*: the precision on majority voted labels.

Methods	RT Prec		Yelp Prec		IMDB Prec	
	avg	voted	avg	voted	avg	voted
Bigram	0.5903	0.6176	0.8045	0.8155	0.7708	0.7609
MTSA	0.8413	0.8783	0.9027	0.8925	0.8921	0.9081

Table 5: Learned Shifters in Yelp dataset ranked by shifter effect values.

Negation	never: -1.33, not so: -1.00, not even: -0.75, not: -0.52, not very: -0.48, not really: -0.39, none: -0.27, no: -0.22, only: -0.18, not that: -0.13, nothing really: -0.11
Diminisher	could: 0.12, reasonably: 0.17, few: 0.18, slightly: 0.18, nothing that: 0.18, felt: 0.22, before: 0.22, not overly: 0.25, would only: 0.25, than: 0.27, somehow: 0.28
Intensifiers	completely: 2.59, more than: 2.42, absolutely: 2.33, extremely: 2.33, really: 2.25, not only: 2.23, some really: 2.17, far: 2.15, particularly: 2.13, simply: 2.12, too: 2.06, excessively: 2.02, certainly: 2.00, most: 2.00, very: 1.96

on sentiment polarity are consistent with our motivation and human intuition. We also analyze the typical errors.

With default settings, MTSA extracts and quantifies around 50 unigram shifters and 20 phrase shifters for RT, 70 unigram shifters and 80 phrase shifters for Yelp, 70 unigram shifters and 100 phrase shifters for IMDB, where the exact numbers varies under different parameter settings. As an example, the top ranked shifters including both unigram and bigram learned from Yelp reviews are presented in Table 5.

4.3.1 Human Judge Evaluation

For each dataset, 200 reviews containing identified shifters are uniformly sampled as an evaluation set. In each review, we rank the sentiment words modified by discovered shifters by their polarity magnitudes. The top-5 sentiment words are marked together with their shifters as a phrase, and are presented with their corresponding polarity scores after rectification, e.g., (“not happy”, -1). Human judges are responsible to evaluate the correctness of polarity scores within review contexts. The baseline for comparison gets the polarity scores from the weights in the bigram logistic regression model if the phrase is there; otherwise, we do not present it. Human judges label the phrases using “correct”, “incorrect” and “irrelevant to sentiment”.

We report intraclass correlation coefficient (ICC) of 4 human judges, the two-way absolute agreement for RT, Yelp, and IMDB respectively: average $k = 0.81, 0.91, \text{ and } 0.92$; individual = 0.68, 0.70, and 0.73. We use majority voting for evaluation, so average measurement is more important. (> 0.8 indicates almost perfect agreement [28]), the ICC shows the judges have achieved agreements in most cases. Since we only inspect the top-5 set and have no idea about the full set, we evaluate the results by the precision, which is calculated by excluding the irrelevant cases (i.e., $Prec = \frac{N_{correct}}{N_{correct} + N_{incorrect}}$), as shown in Table 4.

4.3.2 Shifter Quality Analysis

To analyze the quality of learned shifters, we examine different types of errors made by two different models, bigram model and our sentiment shifter model. The feature space of bigram model is much larger than our sentiment shifter model. First, the typical error of the bigram baseline results from overfitting, e.g., “so great” has unexpected negative sentiment score due to limitation of training data. Nevertheless, our model learned “so” as an intensifier with an effect 1.92, thus

Figure 5: Sentiment polarities in different themes

making the polarity of “so great” even more positive. Second, the typical error our sentiment shifter model made is that the shifter and the sentiment word compose an “unusual” meaning, where the effect deviates from its usual value. Taking “less comfortable” as an example, in our model, “comfortable” holds a positive polarity as an individual word and “less” plays a role of diminisher; although the shifted polarity of the phrase “less comfortable” is less positive than before (i.e., score is lower, but still positive), it should have a negative polarity in the comparison context. Although the deviation results from simplified model, most minor deviations still guarantee the polarity shifting in correct direction. Such contradictions to the general functions of shifters lead to the major errors in our sentiment shifter model and a reasonable treatment to alleviate the issue is to append frequent abnormal phrases into review features.

4.4 Case Study and Discussion

We conduct case studies mainly on Yelp dataset, because of the availability of rich meta-information.

4.4.1 Multi-Theme Assumption Justification

To justify our Assumption 1 that the same word may have different (degrees of) polarities across different themes, we present the sentiment polarities of “cozy”, “cash”, “prepared”, “boring”, “cheap” and “old” learned from Yelp dataset variant, where the theme-descriptors are constructed by categories of business in metadata. The word “cozy” ranks very high in the restaurant category because people usually express positive opinion when a restaurant offers pleasant environment, whereas “boring” appears more negative in the bars category because people enjoy the intriguing environment. The term “cash” has a positive polarity in the restaurant category because many small restaurants prefer cash to credit cards and paying with cash usually has discounts, whereas in other categories, “cash” turns out to be slightly negative.

4.4.2 Explainable Sentiment Analysis

In an ideal scenario, the usage of MTSA should be: 1) train the model on the reviews with sentiment-coherent content in each review and relative confident theme descriptors; 2) apply the model on the sentences or snippets with detected themes. In this way, the model will be able to present: the word-level sentiment polarities and how each word contributes to the overall sentiment by detecting sentiment shifting. We develop an online inference module to both analyze input sentences and offer reasoning. As the case in Figure 6, applying the

Figure 6: Case Study: Explainable Sentiment Analysis

model we trained from Yelp dataset variant, “*spicy*” is an adjective, which is learned to be positive for food, following a learned shifter phrase “*not really*” with negation effect -0.24 and the overall sentiment is reversed to negative. Meanwhile, to justify that MTSA is able to capture *multi-theme* and *sentiment shifting* phenomena, we synthetically create a dataset that include the examples in Figure 1 and corresponding theme descriptors are manually assigned as binary indicators.

4.4.3 Shifters Benefit Discussion

According to our experiments, although we effectively identified and quantified shifters, the sentiment classification accuracy is not substantially improved for long reviews. The limited coverage of shifters is the major reason for the insignificant rectification of shifters for long reviews. From statistical perspective, despite over 93% coverage of rectified reviews, the portion of active features (i.e., sentiment words) adjusted in each review are 7.2/87 in Yelp dataset and 10.5/122.8 in IMDB dataset. From semantic perspective, especially for long reviews, people may express the same opinion more than once in the same review using both acronyms and shifters, (e.g., “*not good*” and “*bad*” in a review). Therefore, shifters may not play important roles for long document classification, but for shorter text or sentence level, they will be more effective.

5. RELATED WORK

Theme-based Review Analysis. Regarding theme sentiment analysis or aspect-based opinion mining [15], previous studies handle the problem either from sentence perspective or document perspective.

Sentence-level analysis tries to build a pipeline to extract theme words, detect themes for each sentence, and classify the polarity of sentence, where themes are often pre-defined [26].

Document-level analysis, such as Latent Aspect Rating Analysis, emphasizes more on extracting major aspects of reviews and infer the latent aspect ratings from each review [38, 33, 34]. These studies focus on aspect rating regression and aspect weight learning for each review as a macroscopic analysis. However, in such models, each word in the vocabulary has a fixed sentiment value for a particular theme and such value is completely independent among themes.

The studies of topic modeling [1] model the text data as a bag of tokens generated from a designed admixture of language models (multinomial distributions over words) and offer a perspective to resolve the multi-theme challenge.

Furthermore, specialized topic models for sentiment analysis like [20, 14, 7], model that sentiment words are generated from either separated sentiment language models or theme-oriented ones. Another way to leverage topic modeling is to prepare the theme assignment to terms. Different from general sentiment lexicon learning, after knowing aspects of terms, Lu et al. [17] learn theme-dependent lexicon. In this work, we adopt the topic modeling as our preprocessing method to generate theme descriptors.

Contextual Valance Shifters for Sentiment Analysis. The contextual valance shifters, illustrated in [25], are actively studied by NLP community. The most intuitive and popular usage of shifters is incorporating shifter-related features into the feature representations of data to improve the sentiment polarity classification. Many of studies utilize simple or linguistic rules to design shifter-related features by discussing both the function of shifters and their scope of affection [9, 37]. However, we found blindly applying static effects of shifters is questionable and has insignificant improvement of classification accuracy.

To automatically discover shifters from data, most existing studies rely on syntactic structures of sentences. In [10], a mutual information based scoring function is used to achieve polarity reversing construction, similar to negation shifters. A rule-based automatic approach for shifter extraction proposed by [2] can only identify the shifters with fixed categories via heuristic rules and is not designed for robust classification. None of them can quantify the effects of shifters.

[5] assumes that the shifting effect is not restricted to shifters but is dependent on the entire sentence, and it adopts SVM classifiers to judge whether a term is shifted in the sentence, whereas [13] proposes an algorithm to automatically generate shifting data. However, the former relies on domain knowledge of sentiment dictionary and the latter turns out to ensemble heterogeneous classifiers.

General Text Classification. Latent representation learning for documents is popular for text classification, from perspectives of topic [1], latent keyphrase [16], and paragraph embedding [12] which extends from word embedding, distributed representations of words [21]. However, instead of word-level analysis, all aforementioned latent representations only offer an overview of the whole document. Deep neural networks show strengths for text classification. The studies of recurrent neural network (RNN) with tree structure [30, 31] and convolution neural network (CNN) [8, 11] for text classification offer an effective direction to model the compositionality for short sequential text data. Although they have shown their superior accuracies for sentiment classification task, in general, they require much more time and resources to train the model and perform inference for prediction. Since tree-based RNN or LSTM requires parsing for sentences, they are not designed for document-level polarity classification. Furthermore, none of them has built a bridge between multi-theme modeling and explicit shifter explanations.

6. CONCLUSIONS

Different from lexicon-based approaches and feature engineering for supervised learning, this work, to the best of our knowledge, is the first to enable both polarity prediction of the same word in reviews with different emphasis on themes as well as discovery and quantification of contextual valance shifters. The data-driven framework, MTSA, is developed to model *multi-theme* and *sentiment shifting* challenges

for sentiment analysis, and let them mutually enhance each other. Our experiments demonstrate effectiveness of MT-SA. In the future, we will extend the learning framework to go beyond bag-of-words feature representations and take advantage of linguistic grammar to distinguish shifters in different linguistic structures to further improve multi-theme sentiment classification and shifter learning.

7. ACKNOWLEDGEMENT

Research was sponsored in part by HP Labs and HPE Vertica, the U.S. Army Research Lab. under Cooperative Agreement No. W911NF-09-2-0053 (NSCTA), National Science Foundation IIS-1320617, IIS-1354329 and IIS 16-18481, HDTRA1-10-1-0120, and grant 1U54GM114838 awarded by NIGMS through funds provided by the trans-NIH Big Data to Knowledge (BD2K) initiative (www.bd2k.nih.gov).

8. REFERENCES

- [1] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993–1022, Mar. 2003.
- [2] N. Boubel, T. François, H. Naets, and I. Cental. Automatic extraction of contextual valence shifters. In *RANLP*, pages 98–104, 2013.
- [3] A. M. Dai and Q. V. Le. Semi-supervised sequence learning. In *NIPS*, pages 3079–3087, 2015.
- [4] J. Han, J. Pei, and Y. Yin. Mining frequent patterns without candidate generation. In *SIGMOD*, pages 1–12, 2000.
- [5] D. Ikeda, H. Takamura, L.-A. Ratinov, and M. Okumura. Learning to shift the polarity of words for sentiment classification. In *IJCNLP*, pages 296–303, 2008.
- [6] L. Jia, C. Yu, and W. Meng. The effect of negation on sentiment analysis and retrieval effectiveness. In *CIKM*, pages 1827–1830, 2009.
- [7] Y. Jo and A. H. Oh. Aspect and sentiment unification model for online review analysis. In *WSDM*, pages 815–824, 2011.
- [8] N. Kalchbrenner, E. Grefenstette, and P. Blunsom. A convolutional neural network for modelling sentences. In *ACL*, pages 655–665, June 2014.
- [9] A. Kennedy and D. Inkpen. Sentiment classification of movie reviews using contextual valence shifters. *Computational intelligence*, 22(2):110–125, 2006.
- [10] W. Kessler and H. Schütze. Classification of inconsistent sentiment words using syntactic constructions. In *COLING*, pages 569–578, 2012.
- [11] Y. Kim. Convolutional neural networks for sentence classification. In *EMNLP*, pages 1746–1751, Doha, Qatar, October 2014.
- [12] Q. V. Le and T. Mikolov. Distributed representations of sentences and documents. In *ICML*, pages 1188–1196, 2014.
- [13] S. Li, S. Y. M. Lee, Y. Chen, C.-R. Huang, and G. Zhou. Sentiment classification and polarity shifting. In *COLING*, pages 635–643, 2010.
- [14] C. Lin and Y. He. Joint sentiment/topic model for sentiment analysis. In *CIKM*, pages 375–384, 2009.
- [15] B. Liu. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, 5(1):1–167, 2012.
- [16] J. Liu, X. Ren, J. Shang, T. Cassidy, C. R. Voss, and J. Han. Representing documents via latent keyphrase inference. In *WWW*, pages 1057–1067, 2016.
- [17] Y. Lu, M. Castellanos, U. Dayal, and C. Zhai. Automatic construction of a context-aware sentiment lexicon: An optimization approach. In *WWW*, pages 347–356, 2011.
- [18] A. L. Maas, R. E. Daly, P. T. Pham, D. Huang, A. Y. Ng, and C. Potts. Learning word vectors for sentiment analysis. In *ACL*, pages 142–150, 2011.
- [19] A. K. McCallum. Mallet: A machine learning for language toolkit. <http://mallet.cs.umass.edu>, 2002.
- [20] Q. Mei, X. Ling, M. Wondra, H. Su, and C. Zhai. Topic sentiment mixture: Modeling facets and opinions in weblogs. In *WWW*, pages 171–180, 2007.
- [21] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean. Distributed representations of words and phrases and their compositionality. In *NIPS*, pages 3111–3119, 2013.
- [22] J. Nocedal. Updating quasi-newton matrices with limited storage. *Mathematics of computation*, 35(151):773–782, 1980.
- [23] B. Pang and L. Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *ACL*, pages 115–124, 2005.
- [24] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up?: Sentiment classification using machine learning techniques. In *EMNLP*, pages 79–86, 2002.
- [25] L. Polanyi and A. Zaenen. Contextual valence shifters. In *Computing Attitude and Affect in Text: Theory and Applications*, volume 20 of *The Information Retrieval Series*, pages 1–10, 2006.
- [26] M. Pontiki, D. Galanis, J. Pavlopoulos, H. Papageorgiou, I. Androutsopoulos, and S. Manandhar. Semeval-2014 task 4: Aspect based sentiment analysis. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 27–35, Dublin, Ireland, August 2014.
- [27] J. Shang, T. Chen, H. Li, Z. Lu, and Y. Yu. A parallel and efficient algorithm for learning to match. In *ICDM*, pages 971–976. IEEE, 2014.
- [28] B. Sharrack, R. A. Hughes, S. Soudain, and G. Dunn. The psychometric properties of clinical rating scales used in multiple sclerosis. *Brain*, 122(1):141–159, 1999.
- [29] R. Socher, B. Huval, C. D. Manning, and A. Y. Ng. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*, pages 1201–1211, 2012.
- [30] R. Socher, A. Perelygin, J. Y. Wu, J. Chuang, C. D. Manning, A. Y. Ng, and C. Potts. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*, volume 1631, page 1642, 2013.
- [31] K. S. Tai, R. Socher, and C. D. Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL*, pages 1556–1566, Beijing, China, July 2015.
- [32] I. Titov and R. T. McDonald. A joint model of text and aspect ratings for sentiment summarization. In *ACL*, pages 308–316, 2008.
- [33] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis on review text data: A rating regression approach. In *SIGKDD*, pages 783–792, 2010.
- [34] H. Wang, Y. Lu, and C. Zhai. Latent aspect rating analysis without aspect keyword supervision. In *SIGKDD*, pages 618–626, 2011.
- [35] S. Wang and C. D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. In *ACL*, pages 90–94, 2012.
- [36] J. Wiebe, T. Wilson, and C. Cardie. Annotating expressions of opinions and emotions in language. *Language resources and evaluation*, 39(2-3):165–210, 2005.
- [37] M. Wiegand, A. Balahur, B. Roth, D. Klakow, and A. Montoyo. A survey on the role of negation in sentiment analysis. In *NeSp-NLP*, pages 60–68, 2010.
- [38] Y. Wu and M. Ester. Flame: A probabilistic model combining aspect based opinion mining and collaborative filtering. In *WSDM*, pages 199–208, New York, NY, USA, 2015.
- [39] X. Yan, J. Guo, Y. Lan, and X. Cheng. A biterm topic model for short texts. In *WWW*, pages 1445–1456, 2013.
- [40] H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 67(2):301–320, 2005.