ABSTRACT

Reviews often contain rich sentiment on multiple themes, where a theme can be an opinion aspect for restaurants or a movie category. Considering the distinctions among themes, the sentiment polarity of a word can be different across themes. Moreover, contextual valence shifters, a family of linguistic patterns, may dramatically change sentiment polarity of contexts that they appear in. Both challenges cannot be modeled effectively 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, due to inconsistency of word sentiment in different themes and sentiment shifting.

To simultaneously resolve the multi-theme and sentiment shifting problems, we propose a data-driven sentiment analysis framework, MTSA, to enable both polarity predictions of the same word in reviews of different themes and, discovery and quantification of contextual valence shifters. MTSA 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. To rigorously formulate the problem, a series of intuitive assumptions are proposed and later verified in extensive experiments conducted on real-world reviews. The improvement of sentiment polarity classification accuracy demonstrates not only the importance of multi-theme and sentiment shifting, but also effectiveness of MTSA. Human evaluations and case studies further show the success of multi-theme word sentiment predictions and automatic effect quantification of contextual valence shifters.

CCS Concepts

•Information systems → Data mining; •Computing methodologies → Natural language processing;

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 and microblogs, has become an important research topic attracting much attention. The polarity classification problem 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 discriminative 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 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][27] 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][26] 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 vice versa. 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 which enable MTSA to examine whether a word has consistent sentiment over different themes, MTSA construct confident training contexts for shifter candidates and derives the shifter effect learning algorithm by modeling shifted sentiments in a logistic regression model. Through bridging two components, iteratively, the embeddings will be further improved by the rectified document feature representations, and vice versa, which enable mutual enhancement.

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 enable 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 \( D \), as well as the sentiment label (polarity or score) \( y_r \) and the theme descriptor \( \Theta \), for each review \( r \in D \), 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 \( D \) and rectify the word-level descriptors \( W \) 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 \( \Theta_r \), where \( \Theta_i \) 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, \( \Theta \in \mathbb{R}^{K \times |D|} \) 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, applying various handcrafted 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 \((i - \Delta_t)\)-th position to \((i + \Delta_t)\)-th position, where \(\Delta_t\) and \(\Delta_r\) 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 outline their effects in the contexts they belong to, we can rectify the word-level descriptors \(W\) and refine the classification model based on the rectified descriptors, as presented in §3.3.

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 extract common sentiment words being invariant across themes, given theme descriptors \(\Theta\) and word-level descriptors \(W\); and (2) the shifter effects learning phase, which learns the valid quantified shifters from candidates and apply them to correct word-level descriptors \(W\), 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 \(W\) accordingly.

3. METHODOLOGY

3.1 Descriptors

**Theme Descriptors.** In this paper, we adopt a probability 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, \(\theta_{ri} = \Pr(t = i|\mathbf{r})\) describes how much different themes are emphasized in the review \(\mathbf{r}\). 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 \(W_r \in \mathbb{R}^{[\Sigma]}\) 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 \(W\) 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 \(W\). 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 \(\delta(r, j) = \Delta t f(r) \times idf(j)\), where \(\Delta t f(r)\) denotes the normalized increment of term frequency in review \(r\) and \(idf(j)\) denotes inverse document frequency of word \(j\) in the corpus \(\mathcal{D}\).

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 \(p_i, q_j \in \mathbb{R}^d\) and defining the sentiment polarity for word \(j\) in theme \(i\) as:

\[
s_{ij} = p_i^T q_j \quad (1)
\]

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

\[
s_r = \sum_{i=1}^{K} \theta_{ri} \sum_{j=1}^{\vert \Sigma \vert} W_{rj} s_{ij} = \sum_{i=1}^{K} \sum_{j=1}^{\vert \Sigma \vert} (\theta_{ri} p_i)^T (W_{rj} q_j) \quad (2)
\]

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

\[
s_r = (P \Theta_r)^T (Q W_r) \quad (3)
\]

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:

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

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:

\[
\mathcal{O} = \mathcal{L} + \Omega(P) + \Omega(Q) \quad (5)
\]

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

Due to the limit of space, we do not derive the optimization algorithm that learns \(P\) and \(Q\) through minimizing the
objective $O$. 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 $D$, 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 $w_1, \ldots, w_m$, its shifted polarity, $s_i' = s_i \prod_{k=1}^{m} f_{w_k}$, where its original polarity is $s_i = \sum_{j=1}^{K} \theta_{i,j} s_{ij}$.

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 $\Sigma_r$ 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 $v_i$ and $v_j$ in word2vec model, and the contextual similarity, $\text{Sim}(i, j) = \frac{v_i \cdot v_j}{\|v_i\| \cdot \|v_j\|}$.

After running bootstrapping process several times, we finally obtain two candidate sets $\Sigma_n$ (123 potential negations) and $\Sigma_p$ (153 potential intensifiers/diminishers) with manual cleaning for obvious wrong candidates. As our algorithm will finally learn the effects of shifters based on 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 [1] to discover phrase shifters $\Sigma_p$. Because allowing non-consecutive phrases will exponentially increase the complexity of the problem, our method only considers the phrases composed with consecutive unigram shifters as potential phrase shifters. $\Sigma_p$ 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 $\Sigma_n = \Sigma_n \cup \Sigma_r \cup \Sigma_p$.

![Figure 2: Sentiment shifter learning process](image-url)

**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.

According to the Theme-invariant Assumption, considering only common theme-invariant sentiment words, when constructing shifter contexts, enables the algorithm to learn reliable effects of shifters. Thanks to multi-theme sentiment modeling, the ubiquitous sentiment words, such as "amazing" and "awful", are able to be automatically extracted and such set of common sentiment words is denoted as $\Sigma_c$. More specifically, given theme embeddings $P$ and word embeddings $Q$, the common sentiment words in $\Sigma_c$ 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
Then, we take the rectified sentiment $s'_{r,j}$ of word $j$ in review $r$ as:

$$s'_{r,j} = s_r + (f_w - f_w^{l-1}) \times \delta(r,j)(\theta, P)^T q_j$$

where $f_w$ is the frequency of word $w$ in review $r$.

Next, assuming the context of shift $w$ in review $r$ is:

$$x_{rw} = \sum_{j \in C_{rw}} \delta(r,j)(\theta, P)^T q_j$$

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

The rectification problem is formulated as follows.

$$\hat{y}_r = \sigma(f \cdot x_r + \hat{\delta}_r)$$

$$\hat{\delta}_r = s_r - \sum_{w \in \Sigma_r} f_w^{l-1} \sum_{j \in C_w} \delta(r,j)(\theta, P)^T q_j$$

The $l_2$ regularization $\Omega(f) = \alpha \sum_{w \in \Sigma_r}(f_w - 1)^2$ 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 $f$:

$$O_f = \sum_{r=1}^{\mid D \mid} y_r \ln \hat{y}_r + (1 - y_r) \ln(1 - \hat{y}_r) + \alpha \sum_{w \in \Sigma_r}(f_w - 1)^2$$

We utilize the optimal solution of the above logistic regression problem as $f^*$ 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: $f_n$ for negation $\Sigma_n$, $f_i$ for intensifier/diminisher $\Sigma_i$, and $f_p$ for shifter phrases $\Sigma_p$, 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 $W$ 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 $f = 1$) for the computation of $W$, 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 $W$ 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 $O_{r,j}$ contains all occurrences of word $j$ in review $r$, we examine if word $j$ presents in any context of shifters and have:

$$W_{rj} = \sum_{k \in O_{r,j}, w \in \Sigma_r \land w \text{ modifies } k} f_w$$

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

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, $P$ and $Q$, together define the multi-theme sentiment polarity classifiers. Although the theme descriptors $\Theta$ are pre-computed and fixed for reviews, the bag-of-words features $W$ are adjusted once the effects of shifters are updated. In the procedure of rectifying $W$ and training more accurate embeddings $P$ and $Q$ 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 $D$ is $N_p$, the size of the vocabulary is $|\Sigma|$, and the number of reviews is $|D|$. The feature extraction in shifter effect learning...
in Algorithm 1 is $O(N_D)$ as the size of the shifter set $|\Sigma|$ can be viewed as a small constant. The model learning for multi-theme sentiment model takes $O(d(K + |\Sigma| + |D|))$, 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. RELATED WORK

**Theme-based Review Analysis.** Regarding theme sentiment analysis or aspect-based opinion mining [13], 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 [25].

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 [28]. These studies focus on aspect rating regression and aspect weights 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 [17] model the text data as a bag of tokens generated from a designed adixture 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 [26], 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 using simple or linguistic rules to design shifter-related features by discussing both the function of shifters and their scope of affection [9] [37]. However, the improvement of classification accuracy is insignificant and blindly applying static shifters effects is questionable.

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, all latent representations mentioned 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) [5] [11] for text classification offer an effective direction to model the compositionality for short sequential text data. Since tree-based RNN or LSTM requires parsing for sentences, they are not designed for document-level polarity classification. 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. Furthermore, none of them has the bridge towards multi-theme modeling and explicit shifter explanations.

5. 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.

5.1 Datasets and Settings

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

- **Rotten Tomatoes (RT) snippets dataset** [1] 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 times-tamps selected from the Yelp Challenge dataset [4] 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** dataset [4] is introduced in [18] as a benchmark for sentiment analysis, where the positive and negative reviews are balanced. Additional 50,000 unlabeled reviews are not used in our experiment.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>RT</th>
<th>Yelp</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>total instances</td>
<td>10,662</td>
<td>30,471</td>
<td>50,000</td>
</tr>
<tr>
<td>test instances</td>
<td>N/A</td>
<td>8,341</td>
<td>25,000</td>
</tr>
<tr>
<td>avg. review length</td>
<td>22</td>
<td>137</td>
<td>231</td>
</tr>
<tr>
<td>theme descriptor dim</td>
<td>5</td>
<td>13</td>
<td>20</td>
</tr>
<tr>
<td>theme extraction</td>
<td>BTM</td>
<td>LDA</td>
<td>LDA</td>
</tr>
<tr>
<td>evaluation</td>
<td>10-fold CV</td>
<td>test set</td>
<td>test set</td>
</tr>
</tbody>
</table>
Theme Descriptor. In our model MTSA, for Yelp and IMDB with long reviews, the LDA implementation in MALLET [19] is adopted to 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 $K = 5$. Inrequent 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 $p = \alpha + \sum_{i:y_i=1} W_r^i$ and $q = \alpha + \sum_{r:y^r=0} W_r^r$ where $\alpha$ serves as smoothing. [35] advises to use binary feature for $W_r$ and defines the log ratio $r$ for words and NB feature as:

$$r = \log \left( \frac{p}{q} \right), \quad \hat{W}_r = r \odot W_r$$

where $\odot$ indicates element-wise product. Because the log ratio $r$ is applied on all features, if we take absolute value of each element in $r$ to obtain $abs(r)$, it serves as a non-negative idf function which measures how discriminative a feature is. Since $r$ 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 schema 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

<table>
<thead>
<tr>
<th>Algo. &amp; Features</th>
<th>RT</th>
<th>Yelp</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>Linear SVM</td>
<td>76.2</td>
<td>91.87</td>
<td>87.8</td>
</tr>
<tr>
<td>Logistic</td>
<td>76.9</td>
<td>91.90</td>
<td>88.19</td>
</tr>
<tr>
<td>NBSVM</td>
<td>78.1</td>
<td>92.13</td>
<td>88.29</td>
</tr>
<tr>
<td>MTSA</td>
<td>78.0</td>
<td>92.20</td>
<td>88.57</td>
</tr>
<tr>
<td>MTSA(fixed negation)</td>
<td>78.3</td>
<td>92.20</td>
<td>88.48</td>
</tr>
<tr>
<td>MTSA(NB)</td>
<td>78.3</td>
<td>92.52</td>
<td>88.81</td>
</tr>
<tr>
<td>MTSA(shifter)</td>
<td>78.4</td>
<td>92.78</td>
<td>88.82</td>
</tr>
<tr>
<td>MTSA(NB + shifter)</td>
<td>78.8</td>
<td>93.08</td>
<td>88.97</td>
</tr>
</tbody>
</table>

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.

Table 2: Comparison of classification accuracy (%).

5.2 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 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 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% discriminatory 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 3. 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 com-

**Table 3: Comparison with newest deep neural models.**

<table>
<thead>
<tr>
<th>Algo.</th>
<th>RT</th>
<th>Algo.</th>
<th>IMDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>CNN</td>
<td>81.5</td>
<td>Paragraph Vec</td>
<td>12</td>
</tr>
<tr>
<td>SA-LSTM</td>
<td>80.7</td>
<td>SA-LSTM 3</td>
<td>92.76</td>
</tr>
<tr>
<td>MTSA</td>
<td>81.3</td>
<td>MTSA</td>
<td>90.44</td>
</tr>
</tbody>
</table>

Table 4: Precision of sentiment shifting. 

<table>
<thead>
<tr>
<th>Methods</th>
<th>RT Prec avg</th>
<th>Yelp Prec avg</th>
<th>IMDB Prec avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bigram</td>
<td>0.8413</td>
<td>0.8783</td>
<td>0.8921</td>
</tr>
<tr>
<td>MTSA</td>
<td>0.8903</td>
<td>0.9176</td>
<td>0.7069</td>
</tr>
<tr>
<td>CNN</td>
<td>0.8045</td>
<td>0.8155</td>
<td>0.8783</td>
</tr>
</tbody>
</table>

For the long review documents and large training corpora, deep neural models may be able to learn good representations but requires very careful tuning on hyper-parameters and massive parameters to learn, e.g., each training review has a embedding vector to learn [12]. Besides, the feature representations learned by deep neural nets will become vector representations for reviews before feeding into a softmax unit and our multi-theme sentiment model can definitely take such vectorial features as W.

### 5.3 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 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.

#### 5.3.1 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, 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.,
Table 5: Learned Shifters in Yelp dataset ranked by shifter effect values.

<table>
<thead>
<tr>
<th>Shifter Type</th>
<th>Effect Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Negation</td>
<td>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, only: -0.18, not that: -0.13, nothing really: -0.11</td>
</tr>
<tr>
<td>Diminisher</td>
<td>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</td>
</tr>
<tr>
<td>Intensifiers</td>
<td>completely: 2.08, more than: 2.14, 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</td>
</tr>
</tbody>
</table>

5.3.2 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 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.

5.4 Case Study and Discussion

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

5.4.1 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.

5.4.2 Explainable Sentiment Analysis

(a) Online Analysis: Yelp dataset variant

The beef does not really look as spicy as you might like

(b) Synthetic Analysis: Synthetic dataset

5.4.3 Shifters Benefit Discussion

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 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 4 and corresponding theme descriptors are manually assigned as binary indicators.

Figure 5: Sentiment polarities in different themes

Prec = \( \frac{N_{correct}}{N_{correct} + N_{incorrect}} \), as shown in Table 4

Figure 6: Case Study: 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 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 4 and corresponding theme descriptors are manually assigned as binary indicators.

5.4.3 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.
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 valence 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. 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. REFERENCES