Aspect Specific Sentiment Analysis using Hierarchical Deep Learning

Himabindu Lakkaraju
Stanford University
himalv@cs.stanford.edu

Richard Socher
MetaMind
richard@socher.org

Chris Manning
Stanford University
manning@stanford.edu

Abstract

This paper focuses on the problem of aspect-specific sentiment analysis. The goal here is to not only extract aspects of a product or service, but also to identify specific sentiments being expressed about them. Most existing algorithms address this problem by treating aspect extraction and sentiment analysis as separate phases or by enforcing explicit modeling assumptions on how these two phases should overlap and interact. In this paper, we propose a novel approach based on a hierarchical deep learning framework which overcomes the aforementioned drawbacks. We experiment with various models of semantic compositionality within this framework. Experimental results on real world datasets show that the proposed framework outperforms other state-of-the-art techniques. In addition, we also demonstrate how domain adaptation using word vectors can benefit the task of aspect specific sentiment analyis.

1 Introduction

With the increase in user reviews on the web, there has been a huge demand for opinion mining techniques which facilitate effective summarization of huge volumes of opinions. This goal can be achieved by identifying specific aspects of a product or service being reviewed and determining the sentiment expressed about these aspects. This task is popularly referred to as aspect specific sentiment analysis in literature.

In order to illustrate the task at hand, let us consider a text snippet expressing a customer's opinion about a particular beer. *"This beer is tasty and leaves a thick lacing around the glass"* This snippet discusses multiple aspects such as the taste of the beer and its appearance. The review expresses positive sentiments about both the aspects. It is interesting to note that the word "tasty" serves both as an aspect as well as a sentiment word in this case. The phrase "leaves a thick lacing" suggests that the snippet is discussing about the appearance of the beer and usage of "thick lacing" can be attributed to positive sentiment. This example demonstrates the intricacies involved in the task of aspect specific sentiment analysis.

In order to tackle the problem at hand, several approaches ranging from heuristic based methods to sophisticated topic models have been proposed. However, there are two major drawbacks with most of the proposed approaches. Firstly, a chunk of them [6, 12] treat the tasks of aspect extraction and sentiment analysis as two separate phases. The process of interleaving these two phases in a more tightly coupled manner allows us to capture subtle dependencies. Secondly, though there exist approaches which consider joint modeling of aspects and sentiments [8, 7, 17], they constrain the way these phases interleave by making rigid modeling assumptions. In order to address the aforementioned drawbacks, we propose a novel deep learning based framework for solving the problem at hand. The major distinguishing factor of this framework is that the joint modeling of aspects and sentiments is carried out without making strict modeling assumptions about the interleaving of aspect and sentiment extraction phases.

2 Related Work

This work spans two major areas within NLP research namely models of semantic compositionality and aspect specific sentiment analysis.

Aspect specific sentiment analysis The problem of aspect specific sentiment analysis has been of great interest in the past decade because of its practical applicability. [6] formulated this problem and proposed association mining based algorithm to extract product features. Wordnet synsets were used to capture sentiment polarity of words. [12] approached this problem by proposing rule based ontologies. More recently, [1] and [9] proposed models for uncovering parts of reviews which mention specific aspects. [8] proposed sentence level topic models to extract aspects and identifying the sentiment polarity. Though these models account for joint modeling of aspects and sentiments, they make assumptions about the syntax of the words and how the syntax governs if a particular word is an aspect or a sentiment. This is not ideal because there are words that we encounter in real world (such as "tasty") which play a dual role of representing both aspects and sentiments.

Models of semantic compositionality In order to capture the semantic compositionality and interactions between words in a phrase, several approaches have been proposed. [11] modeled word compositions by vector addition, multiplication and other simple combinations of word representations. [18] modeled composition of longer phrases using matrix multiplications. More recently, [13, 14] modeled semantic compositionality of sentences and phrases by leveraging parse trees and associating word vectors and interaction matrices with each word in the given phrase and expressed their combination using non-linear functions. Further, [5] integrated the notion of syntax and semantics by bringing together concepts of compositional vector space semantics and combinatory categorical grammar.

In this work, we leverage the strengths of compositional feature representations in order to address the drawbacks prevalent in current solutions for aspect specific sentiment analysis.

3 Our Approach

The basic idea behind our approach is to learn representations for words (word vectors and matrices) which can explain the aspect-sentiment labels at the phrase level. In order to solve this problem, we propose a hierarchical deep learning framework which comprises of dealing with feature representations corresponding to the words and subsequent parses of the phrases and sentences. All these feature representations finally contribute to an objective function that we solve. We further leverage this objective function to come up with multiple formulations to solve the problem. We discuss each of these steps in greater detail below:

Problem Definition: Given a set of sentences $L = \{l_1, l_2, l_3..\}$, identify aspect - sentiment pairs $\{(a_i^1, s_i^1), (a_i^2, s_i^2)..\}$ present in each sentence l_i.

3.1 Compositional feature representations

This phase involves representing each word using a vector and utilizing the binary parse of the sentences as a framework to combine these vector representations in a bottom up fashion as shown in the Figure 1. Each word is represented using a d-dimensional word vector. These d-dimensional vectors can either be initialized randomly or using pretrained vectors [2]. We experiment with multiple models for combining these vector representations. These representations have been proposed in compositional semantics literature [4, 16, 13, 14]. A brief discussion of these models is presented below for the sake of completeness.

3.1.1 Recursive Neural Network (RNN)

This form of semantic compositionality was proposed in [4, 16]. This model associates a d-dimensional vector with each word. As shown in Figure 1, the vectors for the node in the parse tree are computed bottom up. Recursive neural network model uses the following equations to compute the parent vectors :

$$p_1 = f\left(W\begin{bmatrix} b \\ c \end{bmatrix}\right) \qquad p_2 = f\left(W\begin{bmatrix} a \\ p_1 \end{bmatrix}\right)$$

where p_1 and p_2 are parent vectors, and b and c are leaf nodes as shown in Figure 1. $f = tanh$ is a standard element-wise non-linearity. $W \in R^{d\times 2d}$ is the matrix which will be learnt. These vectors are propagated till the root.

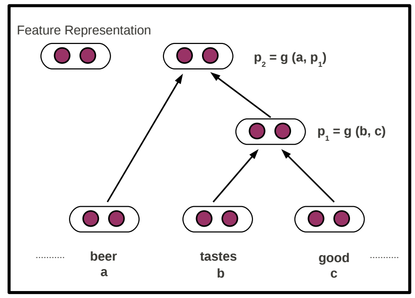

Figure 1: Depiction of the combination of feature representations of words by leveraging the parse of the phrase.

3.1.2 Matrix-Vector RNN (MV-RNN)

This model was introduced in [14]. Each node in the parse tree is associated with a $d \times d$ dimensional matrix and a d-dimensional word vector. This matrix vector representation allows interactions betweeen words to be captured in an elegant way. The matrix representations are initialized using identity matrices with added gaussian noise. Let A, B, C, P_1 and P_2 correspond to the matrix representations of each of the nodes whose vector representations are a, b, c, p_1 and p_2 respectively. This model uses the following equations to compute the parent vectors and matrices :

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right) \qquad P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right)$$

where $W_M \in R^{d\times 2d}$. The vector and matrix representations for the next parent node (p_2, P_2) can be computed in an analogous way. The matrix W is as defined in the RNN model. $f = tanh$ is standard element-wise non-linearity.

3.1.3 Recursive Neural Tensor Network (RNTN)

A challenge with MV-RNN is that the number of matrices and vectors increases linearly with the vocabulary. In order to address this, [15] presented a novel and a more efficient form of semantic compositionality called RNTN.

In this model, each node in the parse tree is associated with a vector and there is no concept of matrices being associated in this model. The interactions are modeled using a tensor which defines multiple bilinear forms. This model uses the following equations to compute parent vectors :

$$p_1 = f\left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]}\begin{bmatrix} b \\ c \end{bmatrix} + W\begin{bmatrix} b \\ c \end{bmatrix}\right)$$

where $V^{[1:d]} \in R^{2d\times d\times d}$ is the tensor defining multiple bilinear forms. Intuitively, each slice of dimensions $2d \times 2d$ in this tensor can be regarded as a compositionality. The vector p_2 can be computed in an analogous manner from vectors p_1 and a.

In summary, we discussed a sequence of increasingly complex compositional feature representations. These models should be seen as *various strategies* that could be plugged into the setup we are proposing[1].

3.2 Objective Function

In this section, we elaborate on extending the representations (discussed in the previous section) to a setting which is meaningful for the task at hand. This can be achieved by setting up an objective

[1]Note that only one of the compositional representations can be plugged in to the objective function

function and tying it to the compositional feature representation appropriately. In order to arrive at this objective function, we begin by posing the problem in a supervised setting.

The main idea behind the approach we employ is that the training process should ensure that the parameters are fit in such a way that the softmax function of the vector representation at the root level of the parse tree $y_i \in C \times 1$ matches the class label of the text snippet as closely as possible. y_i is defined as

$$y_i = \textit{softmax}\,(W_s p_i^{root})$$

where $W_s \in R^{5 \times d}$ is the classification matrix which needs to be estimated and $p_i^{root} \in d \times 1$ is the vector representation at the root level of the parse tree. This is achieved by defining a target distribution vector $t^i \in R^{C \times 1}$. This vector has an entry 1 at the correct label (or labels in case of multi-class classification formulation) and a 0 at other indices.

Our objective is to maximize the probability that the vector representation at the root of each parse tree is as close to the corresponding target distribution vector as possible. This can be achieved by using an objective function which minimizes the cross entropy error between these two vectors. Therefore, the error function to be minimized is given by the equation :

$$E(\theta) = \sum_i \sum_j t_j^i \log y_j^i + \lambda ||\theta||^2$$

Here, θ corresponds to the various parameters of the compositional models we discussed in the previous sub section. The previous section also discussed the computation of the vectors at the root of the parse tree and this vector corresponds to p_i^{root} that we used in the equations above.

3.3 Formulations

In this section, we bring together the concepts of feature representations and objective functions that we outlined previously and discuss in detail how these can be connected to the aspect and sentiment labels of the text snippets. Below, we describe what constitutes class labels and the various ways in which aspects and sentiments can correspond to class labels. Here are the different formulations -

Separate Aspect Sentiment Model (SAS) - In this formulation, we treat aspect extraction and sentiment extraction as two separate phases. We train two separate softmax classifiers, one each for aspect label and sentiment label respectively. In this process, an aspect label and a sentiment label are obtained separately and the (aspect, sentiment) pairs result from the concatenation of the two separate labels. Though this formulation is straightforward and easy to train, it has two major drawbacks. Firstly, as discussed in the introduction, the concept of joint modeling is not facilitated by this formulation. Secondly, this formulation cannot handle the snippets with multiple (aspect, sentiment) pairs because, though it is possible to obtain a chunk of aspect labels and another chunk of sentiment labels (from two separate classifiers), there is no way to associate them appropriately due to the separate training of the two softmax classifiers.

Joint Multi-Aspect Sentiment Model (JMAS) In order to address the shortcomings of SAS, we propose a formulation that trains a single softmax classifier on the aspect-sentiment pairs. The class labels are now aspect-sentiment pairs. For example, (Taste, Positive) corresponds to one class label. This formulation now enables the joint capture of aspects and sentiments elegantly without making any explicit assumptions about their interactions. Further, this model can handle the snippets with multiple (aspect, sentiment) pairs. This can be achieved by allowing more than one element of the target distribution vector t^i (defined in section Objective Function) to be set to the value 1. This set up poses the problem of aspect and sentiment detection as a multi-class softmax classification problem in the context of deep learning.

3.4 Training

All the compositional feature representation models and formulations discussed are trained by computing the gradients of objective function E(θ) with respect to the various parameters. The functions we are dealing with are non-convex and we employ Adagrad optimization procedure [3] to solve the functions. Further, the estimation procedure involves forward computation of vectors and matrices and backpropagation of the appropriate gradients. While backpropagating the gradients, we run into vanishing gradient problems (when gradient values tend to zero) [13, 14].

[13, 14] resolve this problem by propagating the softmax error at the root to all the subsequent levels of the parse tree. However, in our case, this kind of propagation is not ideal since forcing this global softmax error on all the subsequent levels forces the various constituents of a particular text snippet to correspond to the same aspect and sentiment labels as at the root. To illustrate, let us consider the following text snippet *"The beer is very tasty"*. This snippet is associated with the aspect taste and a positive sentiment. (taste, positive) would be the class label at the root (in JMAS formulation). In the case of SAS formulation, the class labels at the roots would be taste and positive respectively. Now, let us consider the constituents of this snippet "The beer" and "very tasty". It would be incorrect if we force the labels at the nodes corresponding to both these snippets to (taste, positive). This is because the phrase "The beer" does not say anything about either the taste or the positive sentiment. This problem can be eliminated if various constituent phrases and words are annotated with appropriate aspect - sentiment pairs. However, annotations at such fine granularities are typically not available in most real world data.

In order to deal with this problem, we use the strategy of propagating the softmax errors from the root only to the initial few levels of the tree. Experimentation revealed that propagating these errors to the initial levels of the parse tree is alleviating the vanishing gradient problem and at the same time, this is not restricting the finer grained constituents of the parse trees to conform to the class labels at the root. We are using the heuristic $\log_2 N$ where N is the number of the levels in the parse tree to determine the number of levels (closer to the root) to which the softmax errors must be propagated. This heuristic worked very well in practice.

4 Experimental Evaluation

In this section, we discuss in detail the experiments carried out to evaluate the proposed framework. We begin with a detailed description of the datasets followed by a discussion on the baselines. Then, we describe the quantitative analysis where we present the results of our models and our experimentation facilitating domain adaptation. Lastly, we conclude this section by discussing the qualitative analysis where we analyze several case based scenarios.

Initialization and Pretraining For all the experiments, the word vectors have been intialized using pretrained vectors from [2]. In case of MV-RNN feature representation, the matrices associated with each word have been initialized as $I + \epsilon$ where I is the identity matrix and ϵ corresponds to gaussian noise.

Dataset Description We used two different datasets for experimental evaluation - beer reviews[2] and camera reviews[3]. The details of these datasets are presented below:

Dataset	# of Sentences	Aspects	Sentiment Scale
Beer Reviews	8532	Aroma, Appearance, Palate, Taste, Beer	1 to 5
Camera Reviews	5008	Price, Battery, Accessories, Display, Portability, Camera	1 to 5

Each sentence in these datasets is labeled with the corresponding aspect - sentiment pairs. The beer review dataset comprises of five different aspects. The camera review dataset comprises of six aspects. In both the datasets, sentiments are expressed on a scale of 1 (highly negative) to 5 (highly positive).

Baselines In order to assess the efficacy of our approach, we compare it against FACTS (FACeT and Sentiment extraction model) and CFACTS (Coherence based FACeT and Sentiment extraction model) models proposed in [8]. FACTS is a generative approach to capture latent facets and associated sentiments. This approach divides words into various syntactic classes and associates a particular syntactic class with aspects and another syntactic class with sentiments. This model represents those classes of approaches which rely on syntactic assumptions for discovering aspects and

[2]http://snap.stanford.edu/data/web-BeerAdvocate.html
[3]http://www.amazon.com/

Approach	Single Aspect - Sentiment Pair (aspect, sentiment) pairs	aspects	sentiments	Multiple Aspect - Sentiment Pairs (aspect, sentiment) pairs	aspects	sentiments
JMAS + RNTN	**66.32%**	**72.02%**	**69.38%**	**69.28%**	**77.04%**	**71.42%**
JMAS + MV-RNN	65.10%	70.23%	69.28%	68.19%	75.48%	69.03%
JMAS + RNN	56.32%	68.92%	58.16%	48.17%	61.11%	52.02%
SAS + RNTN	61.48%	66.78%	63.18%	-	-	-
SAS + MV-RNN	61.29%	67.02%	63.02%	-	-	-
SAS + RNN	52.82%	66.02%	56.91%	-	-	-
Baseline - CFACTS	60.02%	62.33%	60.28%	53.38%	67.31%	53.49%
Baseline - FACTS	59.82%	62.91%	60.02%	52.29%	66.87%	53.01%
Baseline - SVM (tf-idf)	54.38%	66.02%	57.38%	53.92%	64.38%	54.81%
Baseline - NB (tf-idf)	51.97%	63.54%	56.11%	53.36%	62.45%	55.90%

Table 1: Accuracies reported for aspect-specific sentiment analysis - Beer reviews

sentiments. Note that this approach encapsulates the notion of weak coupling between aspects and sentiments via its generative process. On the other hand, CFACTS enforces a stronger dependency between the aspect and sentiment extraction phases via its modeling assumptions. In addition, we also compare our approach against Multi-class Support Vector Machines[4] and Naive Bayes classifiers with tf-idf vectors of words as features.

4.1 Quantitative Analysis

In this subsection, we present the quantitative analysis that we carried out with camera review and beer review datasets in detail.

4.1.1 Single Aspect - Sentiment Pair Detection

In this case, we assume that each text snippet is associated with atmost a single aspect - sentiment pair. We pick only those sentences from our data which are tagged with a single aspect - sentiment pair. There are 8415 sentences in the beer review dataset and 4820 sentences in the camera review dataset which satisfy this criterion. We account for the case where an aspect or sentiment or both may be missing by using the label "empty". So, either the aspect of a sentence or its sentiment or both can be tagged as "empty". The results are presented in Columns 2 - 4 of Tables 1 and 2. The numbers reported are results of a 10-fold cross validation. As can be seen, each of the formulations discussed earlier can be used with various compositionality representations. The tables show various combinations of these. Also, we report three different accuracy numbers - correctness of the prediction of aspect - sentiment pair (Column 2), correctness of the prediction of aspect (Column 3), correctness of the prediction of sentiment (Column 4).

Discussion From Tables 1 and 2, it can be seen that the RNTN and MV-RNN representations outperform RNN representation and other baselines across all the dimensions. This shows that simple concatenation of feature representations of constituent phrases does not work as well as representations where in complex interactions between constituents are allowed. Also, JMAS formulation outperforms SAS formulation which involves independent aspect extraction and sentiment detection phases. This shows that the concept of joint modeling of aspects and sentiments is indeed beneficial. In addition, the baselines CFACTS and FACTS model performs slightly worse than the SAS model. This was mainly due to those data points where aspects and sentiments did not conform to a particular syntactic category. In fact, it is interesting to note that SVM (with tf-idf features) performs aspect detection better than the baseline FACTS model. This is an indication that associating aspects and sentiments with specific syntactic categories might be too constraining in case of the data we are dealing with, where the boundaries between aspect words and sentiment words are blurry and sentiments are more subtle.

4.1.2 Multiple Aspect - Sentiment Pairs Detection

In this case, we relax the assumption that each text snippet should be associated with a single aspect - sentiment pair. In the beer reviews corpus, there are 117 sentences which have multiple aspect

[4] http://www.csie.ntu.edu.tw/ cjlin/libsvmtools/multilabel/

	Single Aspect - Sentiment Pair			Multiple Aspect - Sentiment Pairs		
Approach	(aspect, sentiment) pairs	aspects	sentiments	(aspect, sentiment) pairs	aspects	sentiments
JMAS + RNTN	**73.45%**	**78.10%**	**74.41%**	**75.34%**	**81.02%**	**76.11%**
JMAS + MV-RNN	68.22%	75.28%	71.31%	69.12%	77.48%	72.05%
JMAS + RNN	58.52%	70.11%	63.56%	56.18%	67.87%	58.11%
SAS + RNTN	66.11%	69.14%	68.22%	-	-	-
SAS + MV-RNN	64.81%	69.18%	65.11%	-	-	-
SAS + RNN	53.68%	68.31%	57.91%	-	-	-
Baseline - CFACTS	64.1%	65.65%	65.10%	63.38%	69.44%	65.67%
Baseline - FACTS	61.48%	63.12%	64.19%	62.11%	65.22%	64.19%
Baseline - SVM (tf-idf)	61.11%	67.81%	64.32%	61.12%	66.19%	65.88%
Baseline - NB (tf-idf)	56.19%	61.23%	60.28%	59.08%	63.08%	61.09%

Table 2: Accuracies reported for aspect-specific sentiment analysis - Camera reviews

- sentiment pairs as labels. In the camera reviews dataset, 188 sentences have multiple aspect - sentiment labels. In this part of the experimentation too, we account for absence of aspect or sentiment labels using the class label "empty". A 10-fold cross validation was carried out using only those sentences which had multiple aspect - sentiment labels. In addition, the training set also had all those sentences which had a single aspect - sentiment label. However, the test set solely comprised of those sentences which had multiple aspect - sentiment labels. The results are presented in Columns 5 - 7 of Tables 1 and 2. It can be seen that the entries in these columns corresponding to SAS formulation are empty. This is due to the fact that SAS is tailored towards a single aspect - sentiment label classification.

Discussion Columns 5-7 of Tables 1 and 2 show that RNTN and MV-RNN representations consistently outperform RNN representation and baselines. This indicates that the RNN representation does not capture the interactions between various constituents of sentences well. It is in fact interesting to note that RNN model performs worse than the baselines.

4.1.3 Domain adaptation using word vectors

Another interesting aspect of our analysis constitutes the usage of word vectors obtained from other related datasets as a means of facilitating domain adaptation. We trained the model JMAS + RNTN for the beer dataset by initializing word vectors to those obtained from JMAS + RNTN model for the camera dataset and viceversa. This resulted in an improvement in the sentiment detection by 3.01% and 1.67% in the beer review and camera review datasets respectively. Further, the accuracy of aspect specific sentiment detection increased by 0.87% and 0.83% respectively. Since the domains of beer and camera are not very much related in terms of their aspects, we found that the improvements in sentiment detection did not translate to accuracy improvements of the over all task.

4.2 Qualitative Analysis

In this section, we discuss some anecdotal examples which demonstrate the importance of various concepts crucial to the task of aspect specific sentiment analysis. Through out this section, we refer to the RNTN representations of the respective formulations.

Joint modeling As motivated in the introduction, joint modeling of aspects and sentiments turned out to be important in the process of aspect specific sentiment analysis. We observed several instances in our corpus where clearly the sentiment words were dependent on the aspect under consideration. Similarly, it also seemed that occurrence of certain sentiment words automatically reinforced the presence of related aspects. Amongst all the approaches and their ablations we are dealing with, JMAS concretely enforces this notion of coupling the phases of aspect extraction and sentiment analysis without explicitly constraining the interactions between these phases. On the other hand, SAS does not capture the notion of coupling. Here we examine some sample sentences from the data and their ground truth labels. Then, we discuss how various approaches handled these examples -

- *I'm not getting a huge roasted character which is standard with export stouts, but this is a delicious beer that's highly drinkable* - (Palate, Positive)

- *high carbonation level, kinda thin* - (Palate, Negative)
- *Display quality of the camera is high* - (Display, Positive)
- *This camera is highly expensive* - (Price, Negative)

JMAS formulation correctly identified the aspect - sentiment pairs in each of these cases. However, ablations of SAS failed to capture the sentiment correctly in these examples. The reason being that words such as "high" which are indicative of sentiments in each of these examples have a different meaning based on the aspects they are being associated with. When the word "high" appears alongside "drinkability", it is positive. On the other hand, when it appears alongside "carbonation level", it is negative. Similarly, the word "high" conveys a positive sentiment when it is used to describe "display quality" of the camera. On the other hand, the phrase "highly expensive" indicates a negative sentiment. This nuance could not be captured well by SAS model and whenever words such as "high" whose sentiment was conditioned upon the aspect being discussed appeared, it was interesting to see some sort of a random assignment to sentiment classes. On the other hand, JMAS formulation captured these cases correctly with high probability.

Multiple aspect - sentiment capture We discussed an example in the introduction that clearly highlighted the presence of multiple aspect - sentiment pairs in a text snippet. Here, we present few more such examples (and their ground truth labels) and discuss how well the approaches handled these.

- *This is turning out to be much of the same, with less IPA and more tripel in the smell and taste* - { (Aroma, Positive), (Taste, Positive) }
- *There wasn't any lacing to be seen and for the most part, that was the taste too* - { (Appearance, Negative), (Taste, Negative) }
- *The camera came with a superior quality display, however I am not very convinced if it was worth the money* - { (Display, Positive), (Price, Negative) }

JMAS formulation correctly identified all the aspect - sentiment pairs in each of these cases. SAS formulation is not designed for handling multiple aspects. However, it could predict one aspect - sentiment pair (Display, Positive) of the third example correctly. The predictions of SAS in case of the first and third examples were incorrect.

Relaxing modeling assumptions on interactions between aspects and sentiments The JMAS formulation facilitates joint modeling without explicitly enforcing modeling assumptions on how aspects and sentiments should interact. We observed that this was crucial to the task of aspect specific sentiment analysis. For instance, there were words such as "tasty" which served as indicators of both aspects and sentiments. However, many state-of-the-art approaches (including our baselines) leverage the assumption that aspect words are typically nouns and sentiment words are adjectives. Below we present few examples along with their ground truth labels from our dataset where not having any such assumptions helped in making correct predictions -

- *This is really tasty* - (Taste, Highly Positive)
- *very dark and frothy – no light escapes here at all* - (Appearance, Positive)
- *This is a pricey camera* - (Price, Negative)

All our formulations resulted in correct predictions of aspect-sentiment pairs for all the three examples above. CFACTS and FACTS baselines were unsuccessful in all the three cases.

5 Conclusion

In this work, we attempted to bridge the gap between the literature on semantic compositionality and aspect-specific sentiment analysis. The framework we proposed encapsulates several important modeling decisions, such as joint modeling of aspects and sentiments, the ability to handle the presence of multiple aspects and associated sentiments in a given piece of text, and not making strict modeling assumptions about interleaving aspect and sentiment extraction. The evaluation that we carried out on real-world data demonstrated that our approaches incorporating sophisticated neural semantic composition functions consistently outperform other state-of-the-art techniques, with subsequent qualitative analysis confirming the need for various model elements.

References

[1] Brody, S., and Elhadad, N. 2010. An unsupervised aspect-sentiment model for online reviews. In *HLT-NAACL*.

[2] Collobert, R., and Weston, J. 2008. A unified architecture for natural language processing: deep neural networks with multitask learning. In *ICML*.

[3] Duchi, J.; Hazan, E.; and Singer, Y. 2011. Adaptive subgradient methods for online learning and stochastic optimization. In *JMLR*.

[4] Goller, C., and Kchler, A. 1996. Learning task-dependent distributed representations by backprop- agation through structure. In *ICNN*.

[5] Hermann, K. M., and Blunsom, P. 2013. The role of syntax in vector space models of compositional semantics. In *ACL*.

[6] Hu, M., and Liu, B. 2004. Mining and summarizing customer reviews. In *KDD*.

[7] Jin, W., and Ho, H. H. 2009. A novel lexicalized hmm-based learning framework for web opinion mining. In *ICML*.

[8] Lakkaraju, H.; Bhattacharyya, C.; Bhattacharya, I.; and Merugu, S. 2011. Exploiting coherence in reviews for discovering latent facets and associated sentiments. In *SDM*.

[9] McAuley, J.; Leskovec, J.; and Jurafsky, D. 2012. Learning attitudes and attributes from multi-aspect reviews. In *ICDM*.

[10] Mei, Q.; Ling, X.; Wondra, M.; Su, H.; and Zhai, C. 2007. Topic sentiment mixture: modeling facets and opinions in weblogs. In *WWW*.

[11] Mitchell, J., and Lapata, M. 2010. Composition in distributional models of semantics. In *Cognitive Science, 34(8)*.

[12] Popescu, A.-M., and Etzioni, O. 2005. Extracting product features and opinions from reviews. In *EMNLP*.

[13] Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011a. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *EMNLP*.

[14] Socher, R.; Huval, B.; Manning, C. D.; and Ng, A. Y. 2012. Semantic compositionality through recursive matrix-vector spaces. In *EMNLP*.

[15] Socher, R.; Perelygin, A.; Wu, J. Y.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *EMNLP*.

[16] Socher, R.; Manning, C. D.; and Ng, A. Y. 2011b. Parsing natural scenes and natural language with recursive neural networks. In *ICML*.

[17] Titov, I., and McDonald, R. 2008a. A joint model of text and aspect ratings for sentiment summarization. In *ACL*.

[18] Yessenalina, A., and Cardie, C. 2011. Compositional matrix-space models for sentiment analysis. In *EMNLP*.