Aspect Specific Sentiment Analysis using Hierarchical Deep Learning

Himabindu Lakkaraju, Richard Socher, Christopher Manning
Stanford University

Abstract

This paper focuses on the problem of aspect-specific sentiment analysis, where the goal is to not only extract sentiments, but to understand what aspects of a product or service users are expressing opinions about. 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 an 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 a recently introduced real world data set show that the proposed framework outperforms other state-of-the-art techniques.

Introduction

With the increase in user generated content on the web in the form of customer opinions, there has been a huge demand for opinion mining techniques which facilitate effective summarization of such huge volumes of opinions. While most of the initial work on sentiment analysis involved determining the overall sentiment expressed within a piece of text, drilling deeper and understanding actual reasons of satisfaction and dissatisfaction is more useful. This kind of understanding 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 (Hu and Liu 2004; Popescu and Etzioni 2005) 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. To illustrate, in the example above, the usage of the word "thick" can be interpreted as a positive sentiment only in the context of the aspect appearance and that too when this word occurs alongside an aspect word such as lacing. The word "thick" may refer to a different sentiment when used in the context of another aspect. Such intricacies can be effectively captured via joint modeling of aspects and associated sentiments extraction. Secondly, though there exist approaches which consider joint modeling of aspects and sentiments (Lakkaraju et al. 2011; Jin and Ho 2009; Titov and McDonald 2008a), they constrain the way these phases interleave by making certain modeling assumptions.

In order to address the drawbacks highlighted above, we propose a novel deep learning based framework for solving the problem at hand. This framework essentially facilitates the joint modeling of aspects and sentiments, in addition to modeling the syntactic and semantic dependencies via compositional feature representations. The major distinguishing factor of this framework is that the joint modeling is carried out without making strict modeling assumptions about the interleaving of aspect and sentiment extraction phases. The deep learning framework facilitates the learning of the dependencies between the aspect and sentiment extraction phases without us explicitly encoding it in our model. To the best of our knowledge, this work marks the initial attempt at employing deep learning methodologies for aspect specific sentiment analysis.

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. (Hu and Liu 2004) formulated this problem and proposed association mining based algorithm to extract product features. Wordnet synsets were used to capture sentiment polarity of words. (Popescu and Etzioni 2005) approached this problem by proposing rule based ontologies. (Mei et al. 2007) proposed Topic Sentiment Model (TSM) which jointly models the mixture of topic and sentiment for weblogs. However, this model had the disadvantages of overfitting and inability to handle unseen data.

More recently, (Brody and Elhadad 2010) and (McAuley, Leskovec, and Jurafsky 2012) proposed models for uncovering parts of reviews which mention specific aspects. (Lakkaraju et al. 2011) 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. (Mitchell and Lapata 2010) modeled word compositions by vector addition, multiplication and other simple combinations of word representations. (Yessenalina and Cardie 2011) modeled composition of longer phrases using matrix multiplications. More recently, (Socher et al. 2011a; 2012) 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, (Hermann and Blunsom 2013) 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.

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. This section is divided into appropriate subsections each of which describes one of the aforementioned steps in greater detail.

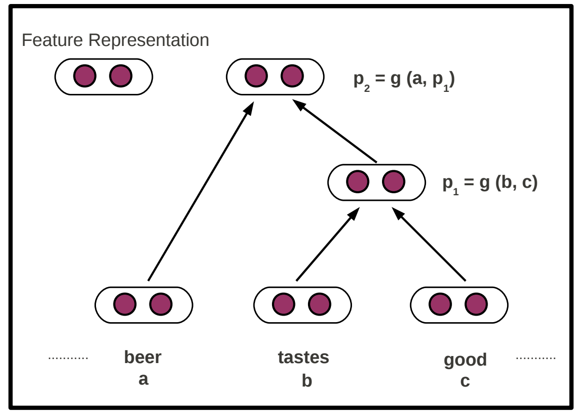

Figure 1: Depiction of the combination of feature representations of words by leveraging the parse of the phrase.

Problem Statement Given a set of text snippets corresponding to opinion expressions $L = \{l_1, l_2, l_3..\}$, identify the set of aspect - sentiment pairs $\{(a_i^1, s_i^1), (a_i^2, s_i^2)..\}$ present in each snippet l_i where the text snippet l_i is a sequence of N_{l_i} words.

Compositional feature representations

This phase involves representing each word using a vector and utilizing the binary parse of the sentence 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 (Collobert and Weston 2008). We experiment with multiple models for combining these vector representations. These representations have been proposed in compositional semantics literature (Goller and Kchler 1996; Socher, Manning, and Ng 2011b; Socher et al. 2011a; 2012). A brief discussion of these models is presented below for the sake of completeness.

Recursive Neural Network (RNN) This form of semantic compositionality was proposed in (Goller and Kchler 1996; Socher, Manning, and Ng 2011b). 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 :

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

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

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.

Matrix-Vector RNN (MV-RNN) This model was introduced in (Socher et al. 2012). 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 :

$$p_1 = f\left(W\begin{bmatrix} Cb \\ Bc \end{bmatrix}\right)$$

$$P_1 = f\left(W_M\begin{bmatrix} B \\ C \end{bmatrix}\right)$$

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.

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, (Socher 2013) 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].

Objective Function

In the previous section, we discussed the methodology of representing phrases and constituent words using compositional vector and matrix forms. Next step involves extending these representations to a setting which is meaningful for the task at hand. This can be achieved by setting up an objective 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.

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

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

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

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.

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.

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 (Duchi, Hazan, and Singer 2011) 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) (Socher et al. 2011a; 2012).

(Socher et al. 2011a; 2012) 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.

Aspect Name	# of Sentences
Aroma	1382
Appearance	1627
Palate	1073
Taste	2336
Beer	2114

Table 1: Aspect Distribution

Sentiment	# of Sentences
Highly negative (1.0 - 2.0)	304
Negative (2.5 - 3.0)	1227
Positive (3.5 - 4.0)	4936
Highly Positive (4.5 - 5.0)	2065

Table 2: Sentiment Distribution

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 dataset we used for the experimentation. This is followed by a discussion on the baselines. Then, we describe the quantitative analysis where we present the accuracy results of our framework along with various ablations and baselines. Lastly, we conclude this section by discussing the qualitative analysis where we analyze several case based scenarios and discuss how various approaches perform in each of these scenarios.

Initialization and Pretraining For all the experiments, the word vectors have been intialized using pretrained vectors from (Collobert and Weston 2008). 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 In order to solve the problem at hand, we used a dataset of 8532 sentences extracted from beer reviews corpus[2]. Each of these sentences is labeled with the corresponding aspect - sentiment pairs. The sentences in this dataset discuss about four different aspects - aroma, appearance, palate and taste. In addition to these four aspects, there are sentences which discuss the topic 'beer'. The sentiments are divided into four different scales - highly negative, negative, positive and highly positive. The data distribution for the aspects and sentiments are given in the Tables 1 and 2 respectively. Note that there are about 117 sentences in this corpus which are labeled with multiple aspect - sentiment pairs. Rest of the corpus comprises of sentences labeled with a single aspect - sentiment pair.

Baselines In order to assess the efficacy of our approach, we compare it against models called FACTS (FACeT and Sentiment extraction model) and CFACTS (Coherence based FACeT and Sentiment extraction model) proposed in (Lakkaraju et al. 2011). 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

[2]http://snap.stanford.edu/data/web-BeerAdvocate.html

syntactic class with sentiments. This model represents those classes of approaches which rely on syntactic assumptions for discovering aspects and 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[3] and Naive Bayes classifiers with tf-idf vectors of words as features.

Quantitative Analysis

In this subsection, we present in detail our analysis on two different experiments we carried out.
Single aspect - sentiment pair detection task: The assumption in this case is that each text snippet is associated with a single aspect-sentiment pair.
Multiple aspect - sentiment pair detection task: This task relaxes the assumption above, thus allowing the presence of multiple aspect-sentiment pairs in a given text snippet.

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. The number of sentences which satisfy this criterion are 8415. We account for the case where an aspect or sentiment or both may be missing by using the label "empty". So, either the aspect or sentiment or both values can be tagged as "empty". The results are presented in Columns 2 - 4 of Table 3. 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 table shows 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 Table 3, 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.

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 our corpus, there are 117 sentences which have multiple aspect - sentiment pairs as labels. In this part of the experimentation too, we account for absence of aspect or sentiment labels using the class label "empty". We trained the model using 8432 sentences which comprise of about 17 multiple aspect - sentiment labeled ones. Remaining sentences in the training set have single aspect - sentiment pair as a label. The test set comprises of 100 sentences each of which is labeled with multiple aspect - sentiment pairs. The results are presented in Columns 5 - 7 of Table 3. 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 Table 3 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 too. Further, all the baseline models exhibit comparable performance.

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 -

- *Drinkability is high* - (Beer, Positive)
- *I'm not getting a huge roasted character which is standard with export stouts, but this is a delicious beer that's highly drinkable* - (Beer, Positive)
- *high carbonation level, kinda thin* - (Palate, Negative)

[3] 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	**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 3: Accuracies reported for aspect-specific sentiment analysis

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. 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) }
- *Smells of roasted malts and mouthfeel is quite strong in the sense that you can get a good taste of it before you even swallow* - { (Aroma, Very Positive), (Taste, Very Positive) }
- *There wasn't any lacing to be seen and for the most part, that was the taste too* - { (Appearance, Negative), (Taste, 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 (Taste, Very Positive) of the second example correctly. The predictions of SAS in case of the first and third examples were incorrect. It is easy to see that sentiments in the first and third examples are more subtle.

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)
- *Pours a clear yellow/gold brew* - (Appearance, Positive)

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.

Conclusion

In this work, we attempted to bridge the gap between the literature on semantic compositionality and aspect-specific sentiment analysis systems. We pointed out important modeling decisions, such as the need for 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. In particular, we proposed a deep learning based framework which can incorporates all these desiderata. Evaluating the proposed framework on a real-world corpus of reviews which carry subtle references to aspects and sentiments, we found that our approaches incorporating sophisticated neural semantic composition functions consistently outperformed other state-of-the-art techniques, with subsequent qualitative analysis confirming the need for our various model elements.

References

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

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

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

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

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

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

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

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

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

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

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

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

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*.

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

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

Socher, R. 2013. Recursive models for semantic compositionality over a sentiment treebank. In *EMNLP*.

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

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