An approach for Aspect Based Sentiment Analysis using Deep Learning

Final Report
CS 585: Introduction of Natural Language Processing

Satya Narayan Shukla, Utkarsh Srivastava
satyanarayan@umass.edu, usrivastava@umass.edu

UMass Amherst
CICS
Fall 2016

Contents

1 Abstract

Sentiment Analysis has been one of the traditional NLP tasks with its applications spanning multiple fields of study. Not only does this project extend its application in finding individual aspect-based sentiments in the domain of user reviews, but also exploits the advantage of using a novel architecture comprised of a Convolutional Neural Network (CNN) to tackle the problem at hand. The inspiration for this project comes from Task-12 of SemEval 2015, titled : "Aspect-Based Sentiment Analysis (ABSA)". The results found through our experiments using the above mentioned deep learning approach proves to work considerably well when compared to the results of the winning teams of the task.

2 Introduction

World economy is a risky system. It is driven by maintaining a balance between the ever-changing consumer behavior and recent trends in the market place. With the advent of the Internet, besides providing a common platform for consumers all around the world to critique and share their opinions on all kinds of products, it has also become a common ground to form holistic opinions to influence a more general crowd with their choices. With a vast amount of data present, analysing it can provide important information to companies and service providers in understanding their customer needs and modifying their service to cater better to the target audience.

Sentiment Analysis is the task of identifying and classifying opinions as positive, negative or neutral. It finds its applications in multiple domains including the identification of political standpoints and brand perception among other fields of study. This makes it an important area of research in both professional and academic setting. For the current project, this study is being applied in the area of direct consumer-product perception by identifying a cohesive sentiment associated with a product as stated by a group of customers in terms of reviews.

This project draws inspiration from Task 12 of SemEval'15 : 'Aspect-Based Sentiment Analysis (ABSA)' by extending the traditional approach of identifying one sentiment per review to finding one sentiment per aspect in each review. It is usually the case that assigning one sentiment to a given review may not capture the intrinsic property of each feature being described for the product at hand. For example, in the context of a restaurant review, one might state the following - "The service was fantastic but the food was kind of awful". Assigning a single sentiment of positive or negative will take away from the fact that the reviewer did like the service aspect ('positive') but did not like the food aspect much ('negative'). Additionally, it may be the case that two separate reviews that have the same overall sentiment may have contradictory opinions about the same aspect. Hence, specifically focusing on ABSA will not only help service providers in spending their resources by analyzing critical areas with the availability of fine-tuned specifications for each aspect, but also provide customers with an insight into individual strengths and weaknesses of the given subject to make a well-formed decision.

The provided data-set from SemEval '15 Task 12 has a separate training and test set provided with data from 'Laptop' and 'Restaurant' domains. Each review is tied with a bunch of Aspect+Polarity tags. In trying to implement a novel approach to tackle the problem, this project takes advantage of a deep learning model like Convolutional Neural Network (CNN) to better understand and reason with the provided data. Two separate models have been built - one to infer about only aspects and the

other to infer about both aspects along with their associated polarities. Most of the top-performing teams relied on using SVMs and CRFs along with the use of linguistic patterns. With the fact that many deep learning approaches have been used for Sentiment Analysis due to their performance, it would be interesting to find how well does this approach work for the current problem representation. Owing to finding a better approach to overcome the shortcomings persistent with the existing methods (linearity for CRFs and coverage for pattern-based learning), we built a CNN to validate our claims of its relatively better performance.

The major contributions of this project include:

1. Designing a deep learning framework in the form of a CNN for ABSA - SemEval '15 Task 12.
2. Experiment with multiple input feature representations for the deep learning framework.
3. Compare and validate our claim of performance against the top-submissions of this task for Laptop and Restaurant domains.

Section 3 describes various kinds of existing approaches in the domain and addresses the limitations of their application for the current data representation. Section 4 further elaborates on the statistics of the dataset with detailed description. Section 5 describes the proposed solution and network architecture used for training and prediction. Section 6 tabulates the results of each experiment performed on the dataset for either domain. Upon experimentation, we got F1-scores of 58.7% and 53.1% for Restaurant and Laptop domains respectively which proved to be comparable and even beat the top-performing teams of SemEval '15 Task 12. Section 7 deals with the discussions about the results. The documentation ends with Section 8 which discusses the conclusions and limitations of the project besides focusing on future work.

3 Literature Review

Aspect based sentiment analysis was first studied by Hu and Liu [4]. They introduced the distinction between explicit and implicit reviews but they only dealt with explicit aspects by using a set of rules based on statistical observations. This method was later improved by Hu and Liu [3], and Popescu and Etzioni [5]. Hu and Liu [3] proposed a trivial algorithm based on linguistic features to extract aspects present in a sentence. They identified frequent nouns and noun phrases after generating POS tags and used them as features. This method does not work well because it cannot find implicit aspects (which are not noun). Popescu and Etzioni [5] classifies a noun or noun phase as a product features by computing pointwise mutual information between the product class and the noun phrase. They assumed that product class is known in advance.

Ding et al. [6] presents a holistic approach that can accurately infer the semantic orientation of a review word based on the review context. It addresses two major problems with identifying semantic orientations of opinions expressed by reviewers on the features of the product, (1) opinion words whose semantic orientations are context dependent, and (2) aggregating multiple opinion aspects in the same sentence. Furthermore, it also considers implicit opinions building on the previous research works where only explicit opinions (expressed by nouns and adjectives) were considered. But given it uses a set of lexical rules, in the absence of a good coverage of such sets of rules, a lexicon based approach can only present limited accuracy.

More and Ghotkar [7] gives a brief insight by evaluating the strengths and weaknesses of a variety of methods being applied for solving the problem at hand. By drawing a comparison between the 4 types of approaches (i.e. frequency-based, relation-based, supervised learning and topic analysis), it emphasizes on the requirement of either a large volume of data or a collection of large number of relation rules to mine appropriate content. Neither of the methods has generated extremely good results on real-world data. Further, they introduce us with the research challenges presented in the form of finding implicit aspects; mining multiple-aspects from a single sentence; cross-domain application of the same system; and the presence of highly unstructured text as input. This in turn motivates us to find a better alternate model to find improvements in both flexibility and accuracy.

Scaffidi et al. [8] used language model to identify aspects in product reviews. They assumed aspect terms are more frequent in reviews than in general natural language text. Their method suffers from low precision because the extracted aspects are affected by noise. There are some other methods which treat this problem of aspect identification as sequence labelling and use Conditional Random Fields. Such methods have performed very well even across domains [9, 10].

Topic modeling has been widely used for extraction of aspects from product reviews[11, 12]. Wang et al. [13] proposed two semi-supervised methods for extracting product's aspect using seeding words. Employing supervised methods, [14] used seed words to guide topic model to learn topics of special interest to user, while [16, 15] employed seed words to extract related product aspects from product reviews.

Recently, deep convolutional neural networks (CNN) [17, 18, 19] have shown significant improvements in performance over the start-of-the-art methods for several natural language processing tasks. Collobert et al. [17] used word embeddings as the input to the CNN to solve standard NLP problems like names-entity recognition, part of speech tagging and semantic role labeling. [19] provides an insight on how a convolutional neural network can be applied in dealing with sentence classification tasks in NLP. With the correct representation of the feature word-vector as input and slight tuning of model hyper-parameters, CNN provides better results than the conventional methods. Also, learning task-specific vectors through fine-tuning offers further gains in performance. Even the simple CNN model with static word vectors give competitive results as compared to more sophisticated deep learning models utilizing complex pooling schemes [20] or requiring parse trees to be computed beforehand [21]. Poria et al. [22] presents a deep learning approach to aspect based sentiment analysis. It uses a 7-layer deep convolutional neural network to tag each word in the review data as aspect word and non-aspect word. They have also comprehended a small set of grammar constructs to use in combination with neural nets to improve the accuracy. Here, features are word embedding along with their part of speech tags. This work shows that a deep CNN is more efficient than existing approaches for aspect extraction. In this project, we use a simple convolutional network with pre-trained word vectors to identify aspects of products present in a review. Our model does not try to tag each word as aspect or non-aspect word like [22] instead identifies the presence of aspects in the whole review, from the given set of labels.

4 Datasets

Given a review, we aim to predict the sentiment of each aspect mentioned in the text. To work on an already annotated and refined data set, we chose the train-test data set available via SemEval 2015 Task 12 as our main source of data. A very detailed documentation and ease of understanding the data representation helped us in choosing this data set. We chose to move forward with this particular data set over the other options available as it is neither as general as the task from SemEval 2014 nor does it build up on more complex details like the ABSA task from SemEval 2016. We have used 2 different data sets, namely:

- Laptop Review Data (Separate Train and Test data)
- Restaurant Review Data (Separate Train and Test data)

The entire data-set and basic guidelines can be found at : http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools. The data is present in an XML format such that each XML file contains a set of reviews in a certain domain. Further, each review is composed of some text (set of sentences) and each sentence has associated aspect-polarity-pair information. The format looks something like:

```
<Review>
   <sentences>
      <sentence>
         <text> —some sentence— </text>
         <Opinions>
            <Opinion category=<Entity#Attribute> polarity=<polarity>/>
            <Opinion category=<Entity#Attribute> polarity=<polarity>/>
             ...
         </Opinions>
      </sentence>
      <sentence>
         ...
      </sentence>
      <sentence>
         ...
      </sentence>
   </sentences>
</Review>
```

Based on the task at hand, this helps us in describing the following terms:

- Aspect : Each aspect is composed of Entity#Attribute pair. An 'Entity' is a concrete thing being talked about. Whereas, an 'Attribute' is a specific characteristic of that entity.
- Polarity : This is a sentiment associated with each Aspect. It can either be Positive, Negative or Neutral in the current task.

In each data file, there exist 4 kinds of sentences:

- A Sentence with exactly one aspect-sentiment information.

- A Sentence which has multiple aspect-sentiments pair associated with it.
- A Sentence that has no aspect-sentiment information.
- A Sentence that has some aspect-sentiment associated but is 'out of scope' for the current set of aspects being taken into consideration.

This in turn helps us in defining the basic statistics of each Set:

- **Laptop Domain**: Both Train and Test data has been separately provided. In terms of the total no. of Entity#Attribute pairs, there can be a total of 22*7=154 different combinations, i.e. 22 Entities and 7 types of attributes. For e.g.:
  Entity: Laptop, Software, OS, Keyboard, Support, Display, Graphics etc.
  Attribute: General, Price, Quality, Connectivity, Miscellaneous etc.
- **Restaurant Domain**: Both Train and Test data has been separately provided. In terms of the total no. of Entity#Attribute pairs, there can be a total of 6*5=30 different combinations, i.e. 6 Entities and 5 types of attributes. For e.g.:
  Entity: Food, Drinks, Service, Ambience etc.
  Attribute: General, Price, Quality, Style & Options, Miscellaneous.

Table 4.1: SemEval data used for evaluation

Domain#	Training	Test	Aspect Types in Train Data	Aspect Types in Test Data
Laptop	1739	761	81	58
Restaurant	1315	685	13	12

An example from each of the Laptop and Restaurant domain is described below:

```
<sentence id="115:1">
   <text>The Computer itself is a good product but the repair depot stinks.</text>
   <Opinions>
      <Opinion category="LAPTOP#GENERAL" polarity="positive"/>
      <Opinion category="SUPPORT#QUALITY" polarity="negative"/>
   </Opinions>
</sentence>
<sentence id="TM#7:0">
   <text>Excellent food for great prices</text>
   <Opinions>
      <Opinion target="food" category="FOOD#QUALITY" polarity="positive" from="10"
to="14"/>
      <Opinion target="food" category="FOOD#PRICES" polarity="positive" from="10" to="14"/>
   </Opinions>
</sentence>
```

Distribution of the aspects and their frequency for Laptop and Restaurant domains for training data are shown in fig. 4.1 and 4.2.

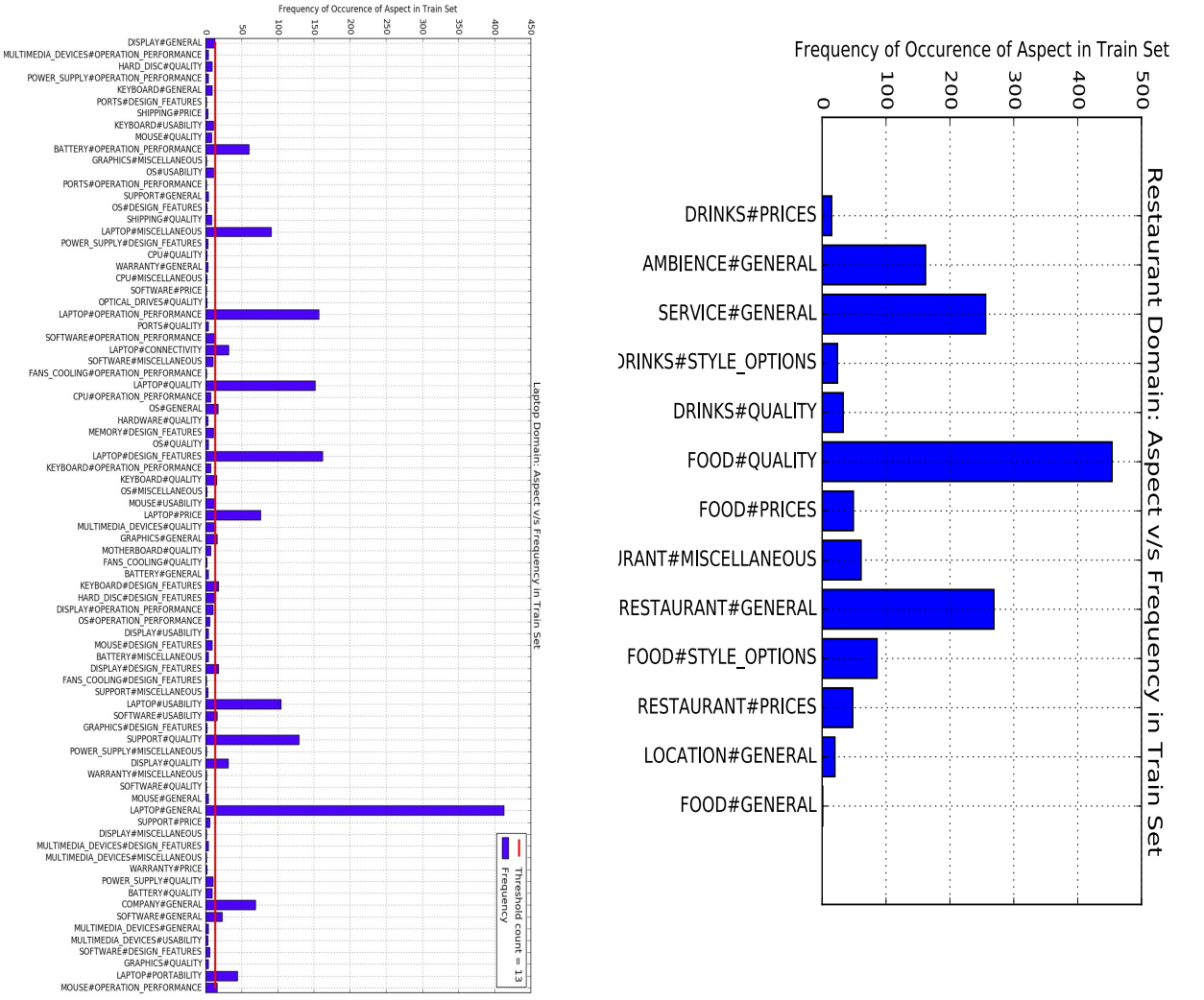

Figure 4.1: Aspect vs Frequency in train set a) Laptop b) Restaurant

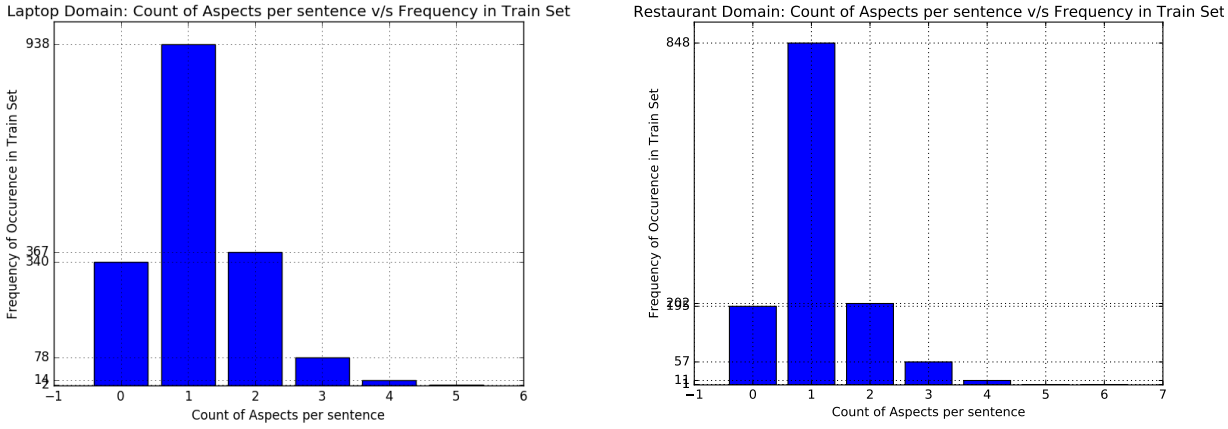

Figure 4.2: Frequency of number of aspects per sentence a) Laptop b) Restaurant

5 Model

Our model (fig. 5.1) consists of an input layer, followed by a convolutional layer, a max-pool layer and then by a fully connected layer with soft-max as our output layer. A sentence matrix $\mathbf{S} \in \mathbb{R}^{|s|*d}$ is built for each input sentence where each row represents a word as a 300-dimensional embedding concatenated with some additional features. The length of sentence |s| is fixed to the maximum length of sentence in the training set, shorter sentences are padded with 0's accordingly to have a fixed input representation. The input layer is $|s| * d$ where d is the dimension of input word representation. For our experiments described in the next section, $|s|$ and d vary. The convolutional layer applies filter to a window of certain number of words in a sentence. The convolution operation between $\mathbf{S}$ and the feature map $\mathbf{F} \in \mathbb{R}^{m*d}$, which slides with a stride of 1 as we want to tag each word, results in a column vector $\mathbf{c} \in \mathbb{R}^{|s|}$. Instead of a single feature map, n feature maps are applied to the sentence matrix resulting in a feature matrix $\mathbf{C} \in \mathbb{R}^{|s|*n}$. We have experimented with different number of feature maps and different sizes of filter matrices. The max pooling layer extracts the maximum value of each column. A hidden layer with h hidden units is applied to the output of the pooling layer. A soft-max layer receives the output of the previous dense layer and calculates the probability distribution over all the possible classes. Since a sentence can contain more than one aspect, we output classes whose probabilities are greater than a threshold value.

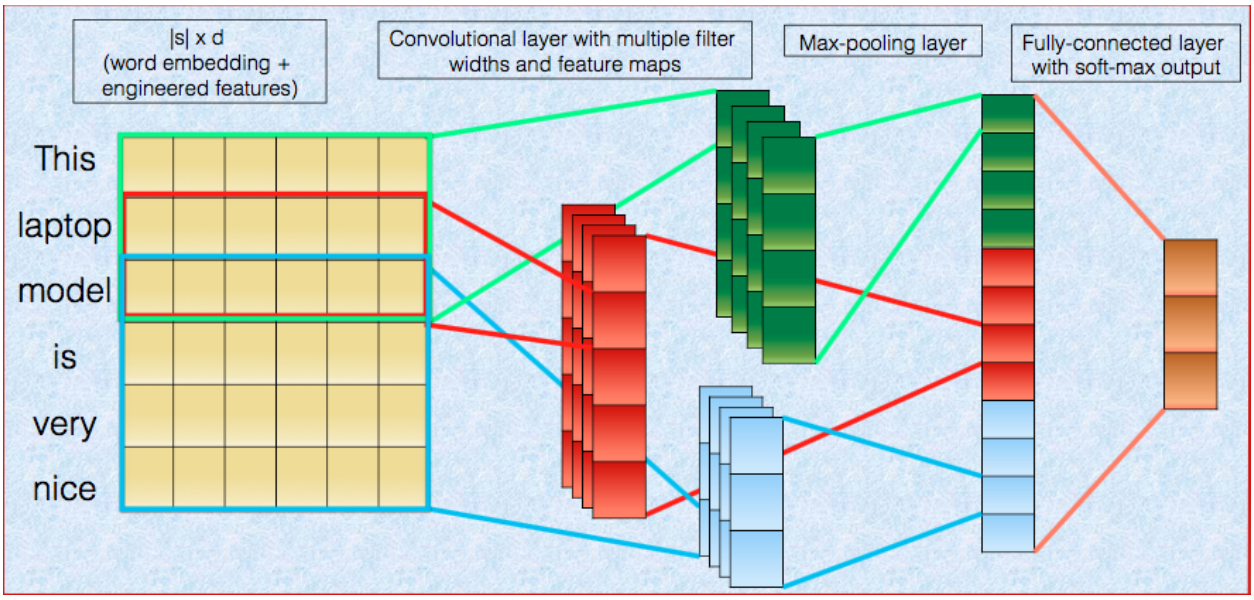

Figure 5.1: Model Architecture (implemented in Lasagne)

We used regularization with 50% dropout on the penultimate (dense) layer with a constraint on L2-norm of the weights. The output of the convolutional layer is computed with hyperbolic tangent function while that of the dense layer output is computed with rectified linear unit (ReLU). The network is trained using Stochastic Gradient Descent algorithm with Nesterov momentum with a learning rate of 0.01 and momentum of 0.9. The categorical cross-entropy is used as the loss function. For identifying the aspects alone, we have created a total of n aspect classes while for predicting aspect and polarity together, this is extended for each aspect-sentiment class so that label $y \in \mathbb{R}^{3n}$ as the polarity can be positive, negative or neutral. Moreover our model also allows to predict multiple aspect-sentiment pairs as there are many such cases in the dataset as shown in fig. 4.2.

6 Experiments and Results

6.1 Preprocessing

After analyzing the frequency of each aspect in the training set as shown in fig. 4.1, we chose to group some of the less frequent tags into one as the training set is small and the network won't be able to correctly classify aspect tags with lower frequency. For Laptop data, we grouped all aspects that occurred fewer than or equal to 13 times as 'OTHER'. Moreover, as a lot of sentences had no aspect tags present at all, so we introduced a new aspect tag 'NONE'. We proceeded to build the network with 23 + 'OTHER' + 'NONE' = 25 output aspect nodes. Also these 23 classes make up for $\sim 85\%$ of the aspect labels in the training data. For Restaurant data, we just introduced a 'NONE' aspect tag to classify the sentences with no aspects. The polarity could be positive, negative or neutral for each aspect except 'NONE'.

Table 6.1: Data after preprocessing

Domain#	Aspect Labels in raw data	Aspect Labels after pre-processing	Aspect#Polarity Labels
Laptop	81	25	73
Restaurant	13	14	40

6.2 Word Embedding

After the sentence is split using 'nltk tokenizer', we use 300-dimensional word embedding to represent each word as a vector. We experimented with publicly available Google and Amazon Embeddings. Google Embeddings are trained on 100 billion word corpus from Google News using continuous bag-of-words architecture [1]. Amazon Embeddings are trained on a large Amazon product review dataset develop by McAuley and Leskovec [2]. This dataset consists of 34,686,770 reviews (4.7 billion words) of 2,441,053 Amazon products from June 1995 to March 2013 and is available at http://sentic.net/AmazonWE.zip. If a token is not found in the word2vec dictionary, the vector is set to 0.

6.3 Hyperparameter Selection

The model implemented requires the following hyperparameters to be experimented upon in finding the optimal value:

1. Filter/Feature Map Sizes
2. No. of each distinct-sized Filter/Feature Map
3. Pool size for Max-pooling layer
4. No. of hidden units in Dense Layer

Additionally, because the output of the model is based on soft-max value, an application dependent threshold value λ needs to be identified based on which the output node will be set to either 1 (found) or 0 (not found). As multiple output nodes can be active for a given input sentence, identifying the correct value of this threshold λ needs to be done.

For experimentation on each choice of unique hyperparameter combination, the training set is split with a ratio of 77:23 into a train-validate set where the model with each set of hyperparameter is

Table 6.2: Optimal Hyperparameter Values

Hyperparamater Type	Optimal Value
Filter/Feature Map Sizes	2 and 3 (Only Aspect) 2, 3 and 4 (Both Aspect+Polarity)
No. of each distinct-sized Filter/Feature Map	100 each for size 2 and 3 (Only Aspect) 100 each for size 2, 3 and 4 (Both Aspect+Polarity)
Pool size for Max-pooling layer	$\|s\|$ i.e extract the maximum value
No. of hidden units in Dense Layer	800

trained on the split train set and examined with validation accuracy on the validation set. Upon experimentation, the optimal set of values received for each hyperparameter can be found in Table 6.2. Besides, tan-h and ReLU have been chosen as activation functions for the Convolutional layer and Dense layer respectively.

For choosing threshold λ, the raw soft-max values from the best performing model are saved and the a graph is plotted to find the value of threshold λ at which the maximum validation accuracy is achieved. As can be seen in the fig. 6.1, a value of λ close to 0.35 performs the best in terms of accuracy for either task.

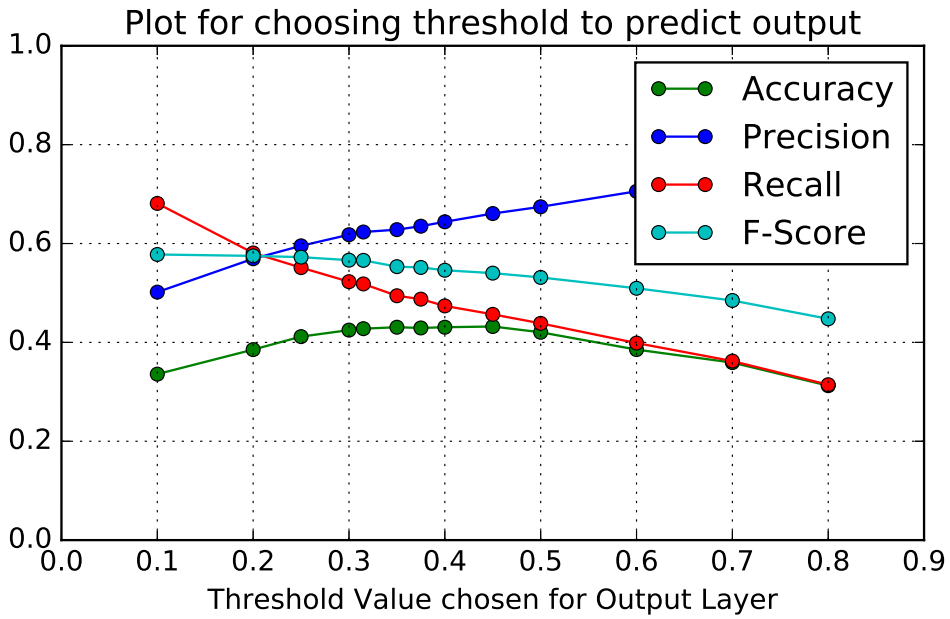

Figure 6.1: Variation of results with threshold value λ (Aspect Model for Restaurant Data)

6.4 Extension of Input Feature Representation - Additional Features (AF)

The input to the model is a $(|s| * d)$ 2-D matrix which represents the word-embedding of each word as individual row vectors. Here $|s|$ is the max word count per sentence among all sentences in the training set and d is the dimension provided by the chosen word-embedding. To provide additional representational capacity to the model, the following have been included:

1. Input size $|s|$ is changed to $|s'|$ such that $|s'| \leq |s|$, where $|s'|$ represents the set of all words that are present in the vocabulary of word embeddings and are not stop words other than negation.
2. A maximum of 11 additional features are concatenated per word with the normal d dimensions to provide specific information about the word.
3. The word-embedding entry for a word is scaled up based on the presence of a specific feature among the 11 added features. This might help for identification as the maximum value to be extracted from the max-pool layer.

For the Aspect Model, 8 out of the 11 features are added, namely:

1. 7 Features to define the POS-tag of the word as provided by the nltk POS-tagger : Noun, Verb, Adjective, Adverb, Preposition, Conjunction and Other. This could help with the fact that generally aspects are Nouns and polarities are Adjectives.
2. 1 Feature to determine the extent of similarity of the current word with a set of defined domain-specific nouns. The Wu-Palmer similarity is used from WordNet with a threshold value of 0.8 to determine the extent of similarity. Also, depending on this being true, the word-embedding for the current word is scaled up. The domain-specific nouns include the correct sense of the following nouns in WordNet:

   (a) Laptop Domain : laptop, OS, software, hardware, keyboard, mouse, battery, charger, memory, brightness, warranty, CPU, graphics etc.

   (b) Restaurant Domain: food, water, dish, menu, restaurant, place, lunch, dinner, breakfast, ambiance, service, host, waiter, bill, quality, pasta, pizza etc.

   For the Aspect+Polarity model, 3 additional features are added, namely:

3. 1 Feature to identify a negation word from the set including but not limited to not, no, don't, couldn't, won't etc. The word-embedding for the current word is scaled up by a lesser extent for this word.
4. 2 Features to denote the extent of closeness with a set of positive words (excellent, good, nice, like, love etc.) and a set of negative words (bad, hate, worse, harsh etc.). This is computed using the cosine similarity between the word-embedding of the current word with each set of provided positive and negative words with a threshold value of 0.8. The word-embedding is again scaled up in the presence on either of this information being consistent.

6.5 Results

For the current task, we have built and trained two CNN models for:

- Aspect Identification and
- Aspect#Polarity Identification

Training each model with optimal hyperparameter values for different kinds of input feature representation, we receive a set of soft-max values for each node at the output layer. After applying the chosen threshold value λ on the output values, we calculate the accuracy of prediction on the unseen test set. The evaluation metric chosen for this task is based on F1-Score which is computed as :

$$\text{F1} - \text{score} = \frac{2PR}{P+R}$$

where P is Precision and R is Recall. Now these values are computed by individually looking at every correct tag present in the gold set instead of computing this as a whole per test sentence. This way even if a statement has correctly predicted 3 out of 4 output tags, it would not mark it entirely as fail but will consider those 3 correctly predicted values in F1-Score computation. The Precision and Recall values are computed against the optimal value of λ found.

Table 6.3 tabulates the Precision, Recall and F1-Score values for each of the 2 models mentioned above against different forms of input representations. Further, figure 6.2 and 6.3 denote the comparison between the F1-score values obtained by our model and the F1-score values of the top-performing teams in Task 12 - SemEval'15. This comparison has been done just for the 'Aspect' CNN model due to the structuring of the competition and associated tasks. The values for comparison have been taken from [23]. Detailed discussion of the results has been described in the next section.

Table 6.3: Precision, Recall and F1-score

Domain	Framework	Aspect			Aspect + Polarity		
		Precision	Recall	F1-score	Precision	Recall	F1-score
Restaurant	Google WE	**65.0 %**	**53.5 %**	**58.7 %**	**53.9 %**	**39.5 %**	**45.6 %**
	Amazon WE	45.1 %	36.8 %	40.5 %	37.1 %	21.6 %	27.3 %
	Google WE + AF	64.9 %	51.7 %	57.5 %	52.1 %	37.2 %	43.4 %
	Amazon WE + AF	49.0 %	37.8 %	42.7 %	36.7 %	28.0 %	31.8 %
Laptop	Google WE	**59.6 %**	**47.9 %**	**53.1 %**	**50.7 %**	**33.7 %**	**40.5 %**
	Google WE + AF	57.6 %	37.4 %	45.4 %	42.6 %	30.2 %	35.3 %

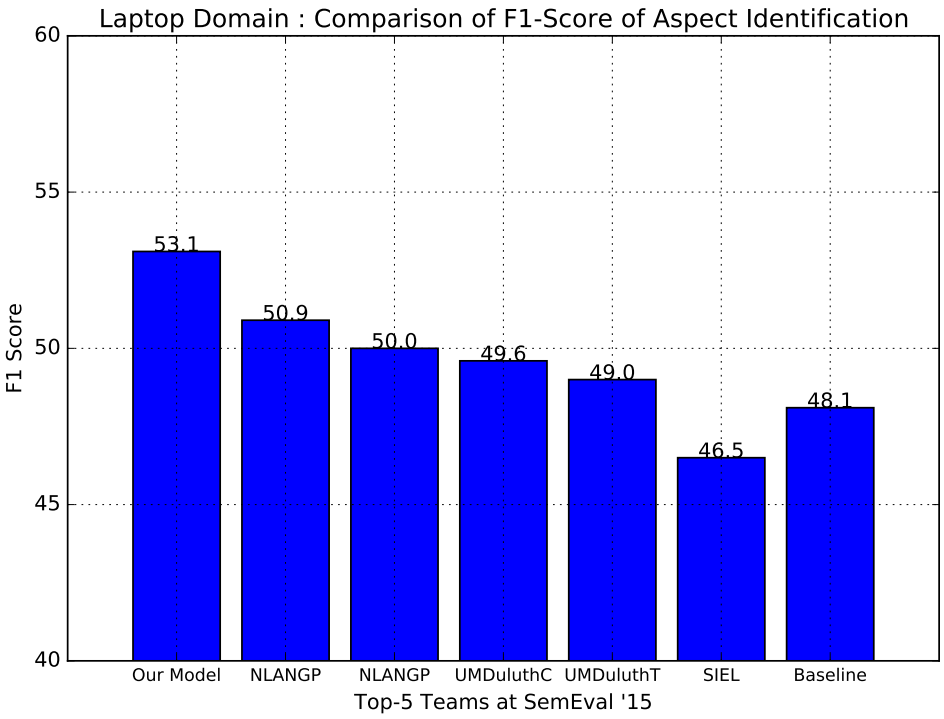

Figure 6.2: Laptop Domain: Comparison of F1-score for Aspect Identification

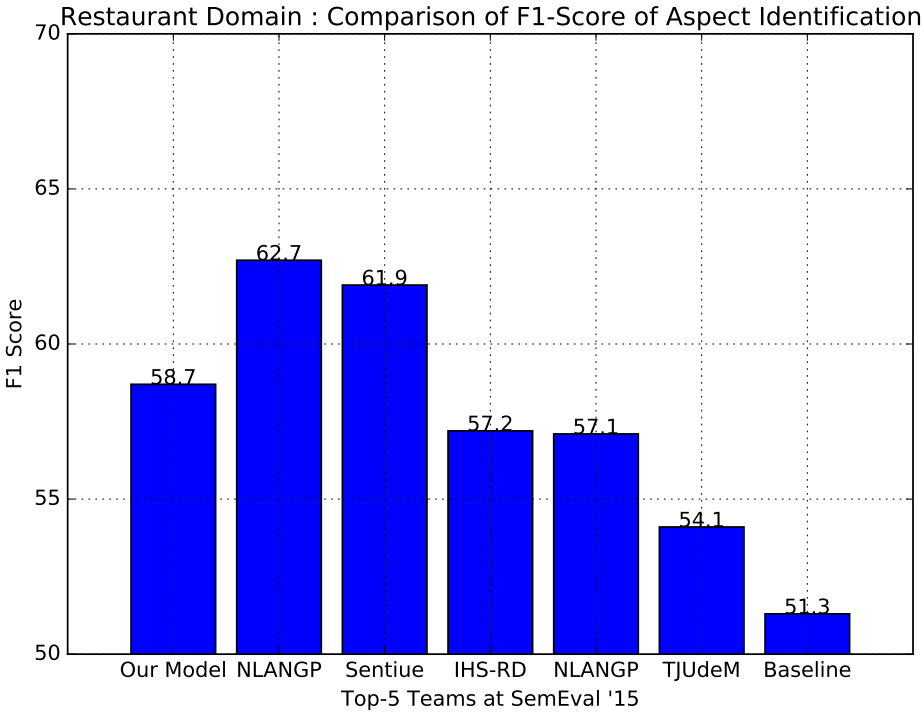

Figure 6.3: Restaurant Domain: Comparison of F1-score for Aspect Identification

7 Discussion

The ABSA Task 12 of SemEval'15 describes 4 kinds of sub-tasks:

1. Aspect Extraction
2. Expression of Aspect in Sentence
3. Polarity Extraction (without being coupled with the associated aspect)
4. Combined (1 & 2)

As stated before, in this project, we have chosen to proceed with sub-task 1 and a modified form of sub-task 3 to couple each identified polarity with associated aspect. Hence, any comparisons done with respect to competition results can be done only for sub-task 1.

As can be seen from Table 6.3, generally tasks associated with the Restaurant domain work better than Laptop domain. This is due to fact that aspects are much more fine-grained in Laptop domain than in Restaurant domain with a total of 81 raw aspect types compared to a total of 13 raw aspect types mentioned for the former and the latter respectively. Hence, given both the training sets are almost of the same size, more data per aspect type is available in the case of Restaurant domain than in Laptop domain. Furthermore, conversion of less frequently occurring aspect types to 'Other' also reduces the evaluation scores slightly.

For comparison in a specific domain, it can be seen that using Google Word-Embeddings without using additional features turns out to be the best. Though, it should be noted that the results of using additional features are very close in comparison. This may be the case because given additional features are based on POS-tagging and word similarities, large sentences may not have the right POS-tagging available and the coverage of words for finding word similarity could be quite limited. Further, it is noted that using Amazon word-embeddings does not provide any improvement in the results when compared to using Google word-embeddings. This could be due to a smaller vocabulary size covered by Amazon word-embeddings when compared to the other option. Additionally, identifying only aspect always provides result values much higher than identifying both aspect+polarity. This is because an extra level of granularity is added when polarity needs to be identified with the aspect as well, leading to a slightly more complex model.

Given that only the structuring of sub-task 1 overlaps with our implementation, we can analyze the results shown in figure 6.2 and 6.3. These results have been taken from [23] which describes a baseline implementation using linear-kernel SVMs based on identifying unigrams from the input sentence. As can be seen, our model performs best in the Laptop domain and performs considerably well for Restaurant domain. The submission from the top team, NLANGP, is based on a multi-class classification problem that has features designed on n-grams, parsing and word-clusters learnt from Amazon and Yelp data. Another team, Sentiue, built a Maximum Entropy classifier with a bag-of-words representation as features. It is clear from the results that the claim of performance of our Deep Learning approach for this task is strongly validated.

Based on the comparison of results for 'Aspect' model, we speculate that extrapolating a similar analysis for the modified sub-task 3 of identifying aspect+polarity together would yield similar performance rankings.

8 Conclusion & Future Work

With the recognition of Aspect-Based Sentiment Analysis as a persistent task in each year's SemEval competition, the need to build better performing systems has become crucial. As the use of deep learning methods is not prevalent in this specific domain yet, in this project, we tried to implement a novel idea by using a Deep Learning approach like CNN to tackle this problem. Based on the different experimentation performed and the final analysis of results against scores from the top-performing teams, we can validate our claim that using deep learning approaches can perform significantly better than the traditional models for the given input representation. With the right kind of feature engineering and model architecture, a deep learning model can exploit the hidden intrinsic structure of the data quite nicely.

Though, apart from searching the optimal set of hyperparameters to use, certain other aspects also need to be taken into consideration which can prove to be challenging in making the system flexible. The choice of threshold value λ for the output layer is application dependent and needs to be chosen carefully upon critical analysis. Also, the input representation should be consistent and well-formed. Words not present in the word-embedding model may define an incomplete model representation in turn leading to degrading the accuracy. Further, dealing with output tags that have very low frequency of occurrence as examples in the training set needs to be properly done. Hence, there exists a trade-off between accuracy value and choosing input representation for such low-count examples.

Though the current implementation suffices to provide good results for this task, a number of improvements can be tried as a part of future work. Given the input word-embedding may not be complete, training the embeddings further on the input training set can provide a good boost in performance for this specific task. Further, extending network layers in the current CNN model or trying out different deep learning approaches like LSTMs or RNNs can be a good starting point to compare strengths and weakness of each kind of approach for such tasks. Also, ABSA task can be extended to predict out-of-domain aspects/polarities for similar domain pairs like Restaurant-Hotel etc. by training on one and predicting on the other. It would be interesting to see if the model is capable of handling such extensions or not.

References

[1] T. Mikolov et al. (2013), “Linguistic regularities in continuous space word representations.” HLT-NAACL, pp. 746-751.

[2] McAuley, J. Leskonev (2013), “Hidden factors and hidden topics: Understanding rating dimensions with review text.” Proceedings of RecSys, Hong Kong, China.

[3] Hu M. and Liu B. (2006), “Opinion extraction and summarization on the web," Proceedings of National Conference On Artificial Intelligence - Volume 2, 1621-1624.

[4] M. Hu and B. Liu (2004), “Mining and summarizing customer reviews.” Proceedings of ACM SIGKDD Conference on Knowledge Discovery & Data Mining, Seattle, pp. 168-177.

[5] Popescu and Etzioni (2005), “Extracting product features and opinions from reviews.” Proceedings of EMNLP, pp. 3-28.

[6] Ding X., Liu B., & Yu P. (2008), “A holistic lexicon-based approach to opinion mining.” Proceedings Of the International Conference On Web Search And Web Data Mining.

[7] Pratima More & Archana Ghotkar (2016), “A Study of Different Approaches to Aspect-based Opinion Mining.” International Journal of Computer Applications 145(6):11-15.

[8] Scaffidi et al. (2007), “Product-feature scoring from reviews.” Proceedings of ACM Conference on Electronic Commerce, pp. 182-191.

[9] Jakob et al. (2010), “Extracting opinion targets in a single and cross-domain setting with conditional random fields.” Proceedings of EMNLP, ACL, pp. 1035-1045.

[10] Zhiqiang et al. (2014), “Aspect term extraction and term polarity classification system.” Proceedings of the International Workshop on Semantic Evaluation, pp. 235-240.

[11] Y. Hu, J. Boyd-Graber, B. Satinoff, A. Smith (2014), “Interactive topic modeling.” Machine Learning 95 (3), pp. 423-469.

[12] Z. Chen, B. Liu (2014), “Mining topics in documents: standing on the shoulders of big data.” Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 1116-1125.

[13] Wang et al. (2014), “Product aspect extraction supervised with online domain knowledge.” Knowledge Based Systems 71, pp. 86-100.

[14] J. Jagarlamudi, H. Daume III, R. Udupa (2012), “Incorporating lexical priors into topic models.” Proceedings of the 13th EACL Conference, Association for Computational Linguistics, pp. 204-213.

[15] A. Mukherjee, B. Liu (2012), “Aspect extraction through semi-supervised modeling.” Proceedings of 50th Annual Meeting of the ACL: Long Papers, Volume 1, ACL, pp. 339-348.

[16] H. Wang, Y. Lu, C. Zhai (2010), “Latent aspect rating analysis on review text data: a rating regression approach. Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, pp. 783-792.

[17] R. Collobert, J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu, P. Kuksa (2011), "Natural language processing (almost) from scratch." Journal of Machine Learning Research 12, pp. 2493-2537.

[18]] S. Poria, E. Cambria, A. Gelbukh (2015), "Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis." EMNLP, pp. 2539-2544.

[19] Kim, Y. (2014), "Convolutional Neural Networks for Sentence Classification." arXiv.org. 1408.5882

[20] N. Kalchbrenner, E. Grefenstette, P. Blunsom (2014), "A Convolutional Neural Network for Modeling Sentences." Proceedings of ACL.

[21] R. Socher, A. Perelygin, J. Wu, J. Chuang, C. Manning, A. Ng, C. Potts (2013), "Recursive Deep Models for Semantic Compositionality Over a Sentiment Treebank." Proceedings of EMNLP.

[22] Poria S., Cambria E., & Gelbukh A. (2016), "Aspect extraction for opinion mining with a deep convolutional neural network." Knowledge-Based Systems, 108, 42-49.

[23] Pontiki, Maria, et al. (2015), "Semeval-2015 task 12: Aspect based sentiment analysis." Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015), Association for Computational Linguistics, Denver, Colorado.