Athens University of Economics and Business

MSc in Computer Science

Master's Thesis

"Aspect Based Sentiment Analysis

on Electronics Reviews in English"

Dimitris Lolis

EY1607

Supervisor: Ion Androutsopoulos

Athens, June 2018

Abstract

In recent years more and more businesses use the feedback mechanism of reviews for their products and services, in order to adjust to the increasing needs of the customers. In order for this task to become more automated and efficient, sentiment detection out of texts (Sentiment Analysis) is of paramount importance.

Riding on the trends of deep learning, this thesis tackles the extraction of the aspects from sentences (e.g. in “This laptop is AMAZING” the aspect of this sentence is the word “laptop”) as well as the sentiment towards them (Aspect Based Sentiment Analysis) with satisfying results. Using SemEval 2016 Task 5 Sub task 1 as a framework for the thesis, we designed two models, for aspect extraction and sentiment detection respectivelly.

For aspect extraction we created a BiLSTM with self-attention in order to predict multiple aspects contained in a sentence, and for Sentiment Detection a Convolutional Neural Network with aspect embeddings that “influenced” the model towards the correct sentiment. The methods used to tackle aspect extraction and aspect sentiment detection are both without any need for expensive feature engineering. Using the SemEval team submissions as baselines, our model surpassed the winner in aspect extraction and came second in Sentiment Detection. In order to achieve higher results than the winners in aspect extraction we used data augmentation as well as word embeddings from Amazon electronics reviews, since SemEval’s original dataset was quite small for a deep neural nets implementation. Finally, we used transfer learning using the Amazon electronics dataset with satisfying results in Sentiment Detection.

Περίληψη

Κατά τη διάρκεια των τελευταίων ετών όλο και περισσότερες επιχειρήσεις χρησιμοποιούν τον μηχανισμό ανατροφοδότησης αξιολογήσεων για τα προϊόντα και τις υπηρεσίες τους, ούτως ώστε να προσαρμοστούν στις αυξανόμενες ανάγκες των καταναλωτών. Για να γίνει πιο αυτοματοποιημένη και αποτελεσματική αυτή η εργασία είναι υψίστης σημασίας η ανίχνευση συναισθήματος από τα κείμενα (Sentiment Analysis).

Ακολουθώντας την τάση της βαθιάς μάθησης, αυτή η διπλωματική εργασία ασχολείται με την εξαγωγή χαρακτηριστικών από τις προτάσεις (π.χ. το χαρακτηριστικό της προτάσεως «Αυτό το λάπτοπ είναι ΕΞΑΙΡΕΤΙΚΟ» είναι η λέξη "λάπτοπ") και τα συναισθήματα αυτών (Aspect Based Sentiment Analysis-Ανάλυση Συναισθήματος Βασισμένη σε Χαρακτηριστικά) με ικανοποιητικά αποτελέσματα. Χρησιμοποιώντας το subtask 1 του task 5 από το SemEval 2016 ως ένα πλαίσιο της διπλωματικής εργασίας, σχεδιάσαμε δύο μοντέλα, ένα για την εξαγωγή χαρακτηριστικών και ένα για την ανίχνευση συναισθήματος.

Για την εξαγωγή συναισθήματος δημιουργήσαμε ένα BiLSTM με αυτο-προσοχή (self-attention) το οποίο προβλέπει πολλαπλά χαρακτηριστικά που μπορεί να περιέχονται σε μια πρόταση. Για την Ανίχνευση Συναισθήματος δημιουργήσαμε ένα Συνελικτικό Νευρωνικό Δίκτυο με ενσωματώσεις χαρακτηριστικών (aspect embeddings), οι οποίες «επηρεάζουν» το μοντέλο προς το σωστό συναίσθημα. Οι μέθοδοι που χρησιμοποιήθηκαν για την εξαγωγή χαρακτηριστικών και την ανίχνευση του συναισθήματος δεν χρήζουν δαπανηρού «feature engineering». Χρησιμοποιώντας τα συστήματα των συμμετεχόντων ομάδων του SemEval ως σημείο αναφοράς, το μοντέλο μας ξεπέρασε το νικητή στην εξαγωγή χαρακτηριστικών και ήρθε δεύτερο στην Ανίχνευση Συναισθήματος. Για να μπορέσει το μοντέλο να επιτύχει καλύτερα αποτελέσματα από τους νικητές στην εξαγωγή συναισθήματος, χρησιμοποιήθηκε επαύξηση δεδομένων (data augmentation) και ενσωματώσεις λέξεων (word embeddings) από κριτικές ηλεκτρονικών προϊόντων του Amazon, καθώς το αρχικό σύνολο δεδομένων του SemEval ήταν σχετικά μικρό για την υλοποίηση ενός βαθιού νευρωνικού δικτύου. Τέλος, χρησιμοποιήσαμε μεταφορά μάθησης (transfer learning) στο σύνολο δεδομένων των ηλεκτρονικών προϊόντων του Amazon με ικανοποιητικά αποτελέσματα στην ανίχνευση συναισθήματος.

Acknowledgements

At this point I would like to thank my supervisor Ion Androutsopoulos for providing me with a cutting edge thesis topic, that combines the very interesting fields of NLP and Machine learning. For this endeavor I received help and a lot of insights from AUEB's NLP Group as well as from John Koutsikakis, George Brokos and Angeliki Karampini. Finally, I would like to thank my family and all the people close to me for their support and encouragement.

Athens 2018

Dimitris Lolis

Contents

Tables

Figures

Chapter 1 – Introduction

1.1 Sentiment Analysis

Sentiment Analysis (SA) is a challenging text mining problem that seeks to identify the underlying sentiment out of natural language. Many companies actively encourage users to submit their thoughts on the product or service that was delivered to them. Using that feedback they can adjust to the needs of the customers and provide a significantly improved user experience. Aspect Based Sentiment Analysis (ABSA) mines texts for opinions about specific entities (e.g., hard disks, laptops, graphics cards), and their attributes (e.g., performance, temperature, price) providing insights to both consumers and businesses.

An ABSA method can analyze large amounts of unstructured texts and extract information not necessarily included in the user ratings that are available in some review sites. For the purposes of this thesis *Aspect Based Sentiment Analysis on Reviews for Electronics* we created two models, using deep learning architectures, for Aspect Extraction and Sentiment Detection respectively. For the training, validation and testing of the models we used the SemEval 2016 Task 5 Subtask 1 (Sentence level ABSA) English datasets for laptops. The SemEval 2016 shared task on ABSA is the continuation of the ABSA tasks of SemEval 2014 and 2015. For 2016, the organizers provided 19 training and 20 testing datasets written in 8 languages, including 7 domains, as well as a common evaluation procedure. From these datasets, 25 were for sentence-level and 14 for text-level ABSA; the latter was introduced for the first time as a subtask in SemEval. The task attracted 245 submissions from 29 teams. By using the resources of the SemEval competition, we were provided with a baseline as well as the results of other submissions, in order to test the competency of the models suggested in this thesis against them [6].

1.2 Tasks Description

The SE-ABSA16 task consisted of 3 Subtasks; participants were free to choose the subtasks, slots, domains and languages they wished to participate in. This thesis got involved only with Subtask 1.

Subtask 1 (SB1): Sentence-level ABSA. Given an opinionated text about a target entity, identify all the opinion tuples with the following types of information:

• AE: Aspect Category Extraction.

Identification of the entity E and attribute A pairs towards which an opinion is expressed in a given sentence. A and E should be chosen from predefined inventories of entity types (e.g., "restaurant", "food") and attribute labels (e.g., "price", "quality").

• ASD: Aspect Sentiment Detection.

Each identified E#A pair has to be assigned one of the following polarity labels: "positive", "negative" or "neutral".

1.3 Datasets

SemEval provided its contestants with a total of 39 datasets in the context of the SE-ABSA 2016 task; 19 for training and 20 for testing. The texts were from 7 domains (Restaurants, Laptops, Mobile Phones, Digital Cameras, Hotels and Museums) in 8 Languages (English, Arabic, Chinese, Dutch, French, Russian, Spanish and Turkish). A total of 70790 manually annotated ABSA tuples were provided for training and testing; 47654 sentence level annotations in 8 languages for 7 domains. The restaurant, hotel, and laptops datasets were annotated at the sentence-level following the respective annotation schemas of SE-ABSA15.[1] For laptops, the sentences were annotated as the following example suggests:

"It is extremely portable and easily connects to WIFI at the library and elsewhere."

→ {cat: "laptop # portability", pol: "positive"}, {cat: "laptop # connectivity", pol: "positive"}.

The laptop data were annotated by 5 undergraduate computer science students. The resulting annotations were then inspected and corrected (if needed) by an expert linguist (annotator A) and one of the task organizers (annotator B). Borderline cases were resolved collaboratively by annotators A and B [6].

[1]http://alt.qcri.org/semeval2015/task12/data/uploads/semeval2015_absa_laptops_annotationguidelines.pdf

1.4 Outline

Chapter 2 – Related Work: This chapter describes various approaches in ABSA related tasks, as well as Sentiment Analysis in general.

Chapter 3 – Aspect Based Sentiment Analysis: This chapter explains in detail the methods used for Aspect Extraction and Aspect Sentiment Detection in this thesis.

Chapter 4 – Experiments: This chapter contains the results of the implemented models.

Chapter 5 – Conclusions: This chapter summarizes the results of the thesis and proposes future work.

Notation

ABSA	Aspect Based Sentiment Analysis
CNN	Convolutional Neural Network
LSTM	Long Short Term Memory
RNN	Recurrent Neural Network
AE/ ASD	Aspect Extraction / Aspect Sentiment Detection
E#A pair	Entity Attribute pair (e.g., LAPTOP#SOUND)
SVM	Support Vector Machines
POS	Part Of Speech
NER	Name Entity Recognition

Table 1- Abbreviations.

Chapter 2 – Related Work

2.1 Aspect Category Detection

In this thesis, related work is focused on ABSA and approaches from the SemEval ABSA task [6]. However, it is noted that there is a plethora of works related to this thesis in a broader manner [5, 13, 14].

First and foremost, we have to note that the organizers of the SemEval 2016 ABSA task provided a trained SVM classifier [6] to serve as a baseline for the contestants. For Aspect Extraction there were several approaches worth noting: Toh and Su [15] implemented a system that consists of multiple binary classifiers that predict the categories of the data. Each classifier is trained using a single layer feed forward network operating, among others, on features neural network features generated by a Deep Convolution Network proposed by Severyn and Moschitti [3]. Xenos et al [4] used multiple ensembles, based on SVM classifiers. The architecture of our model was based heavily on the work of Wang and Liu [5], who used combined models for Aspect Extraction and Sentiment Detection.

2.2 Sentiment Polarity

Aspect Based Sentiment Analysis (ABSA) is traditionally split into an aspect extraction and a sentiment analysis subtask. Similarly to the baseline of aspect extraction we have an SVM classifier provided by the competition to measure the accuracy of the submitted models [6]. Tang et al. [15] used a target-dependent LSTM to determine sentiment towards a target word, while Nguyen and Shirai [11] used an RNN that leverages both constituency as well as dependency trees. Ruder et al. [9] implemented a convolutional neural network based on Collobert et al. [7] both for AE and ASD. Kumar et.al [2] used a system incorporating domain dependency graph features, a distributional thesaurus (See Section 5.2) and unsupervised lexical induction using an unlabeled external corpus for AE and ASD. Xenos et al. [4] used an ensemble of two supervised classifiers, one based on hand crafted features and one based on word embeddings. Khalil and Beltagy [12] used a CNN classifier initialized with fine-tuned word embeddings. Wang et al.

[14] used an attention-based Long Short-Term Memory (LSTM) network for ASD. Finally, Wang and Liu [5] use a Convolutional Neural Network with ReLu nonlinearities.

2.3 Thesis Contribution

Many works report very good results in both AE and ASD, as well as the combination of the two, as seen in Pontiki et al. [6]. In this thesis we tried to use some of the previous methods in our own implementations and put some of their ideas of future work to the test by trying to produce better results using a Deep Learning implementation. We managed to achieve higher micro F1 for Aspect Extraction and accuracy for ASD using data augmentation and Transfer Learning, which will be described in detail in Section 3.4.2, showing that there is room for improvement for these models.

Chapter 3 – Aspect Based Sentiment Analysis

3.1 Description of Aspect Extraction and Sentiment Polarity Detection

ABSA usually consists of AE and ASD, which are the stages that extract the aspects of a sentence and then estimate the sentiment polarity per aspect, respectively. The sentences may contain more than one aspect; e.g. "I bought it for really cheap also and its AMAZING.". This review refers to the aspects LAPTOP#PRICE and LAPTOP#GENERAL and they both have a positive sentiment towards their respective aspect.

Given that the provided laptop dataset contains 81 aspects plus the OTHER aspect, it makes it a multi-label classification task, especially due to the fact that many aspects co-exist in the same sentence. ASD faces another kind of difficulty, since the model will have to detect sentiment polarities for different aspects that may contradict each other.

3.2 Datasets

For the initial experiments we used the dataset[2] provided by the the SemEval ABSA competition. The English laptop dataset is described by the following Figures 1, 2 and Table 2. Due to the fact that the initial results proved to be inadequate to surpass the winning submissions, the method of data augmentation (see Section. 4.3) was put to use.

For the purposes of this thesis the datasets used were:

- Dataset SemEval.
    - The original dataset provided by the competition.
- SemEval#GE.
    - The concatenation of the Original and the German pivoted dataset.
- SemEval#GE+FR.
    - The concatenation of the Original, German and the French pivoted dataset.

[2] http://metashare.ilsp.gr:8080/repository/browse/semeval-2016-absa-laptop-reviews-english-test-data-phase-b-subtask-1/0d164076c0dd11e5bcd5842b2b6a04d775171b8a4e5849609a629e20ab03a8cd/

- SemEval#GE+FR+SP.
    - Same as the SemEval#GE+FR but with the addition of the pivoted Spanish dataset.

All of the above datasets were used to examine whether the implementations for Aspect Extraction and Sentiment Polarity Detection perform better when they have a larger training dataset.

Language	Domain	#Train reviews	#Train Sentences (opinionated)	#Evaluation reviews	#Evaluation sentences
English	Laptops/Electronics	315	1400	135	600

Table 2 - Dataset Statistics.

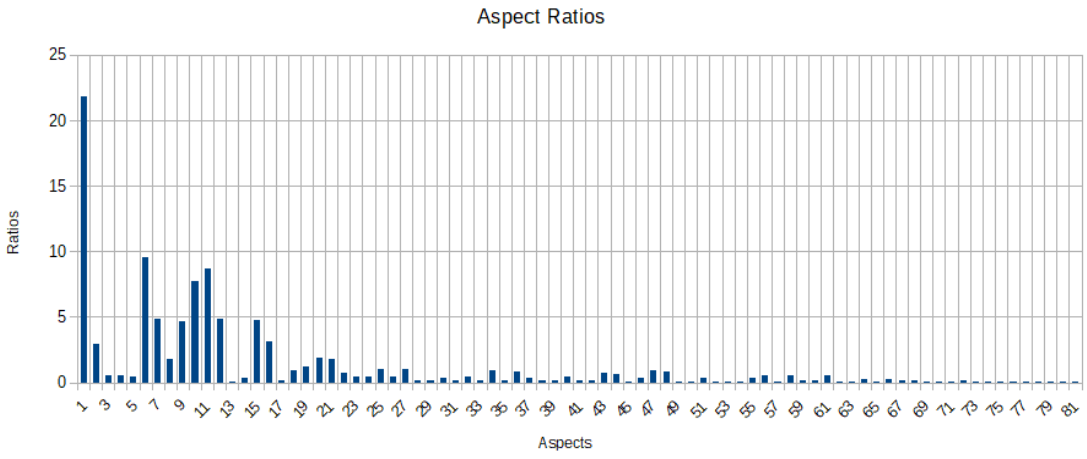

Figure 1 - Aspect Ratios.

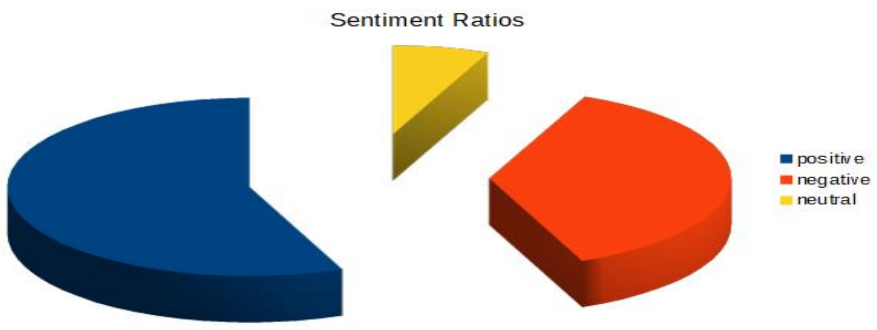

Figure 2 - Sentiment Frequencies.

Word Embeddings

Since the provided dataset was quite small to produce word embeddings from, in order to have a more spot-on representation of our reviews, we decided to create our own word embeddings[3] from Amazon's[4] Electronics dataset, using Gensim Word2Vec.[5] This dataset contained 1.689.188 reviews thus allowing us to create word embeddings from a similar domain. The hyper-parameters that were explicitly used for Gensim were:

- Min count = 5.
- Size = 300.
- Workers = 4.
- CBOW is used as the model.

Gensim accepts further customization for the following hyper-parameters: window, alpha, seed, max_vocab_size, sample, hs, cbow_mean, hashfxn, iter and trim_rule which were left to default values.

[3] https://gitlab.com/dimitrisl/Starting_Area/tree/master/word_embeds
[4] http://snap.stanford.edu/data/amazon/productGraph/categoryFiles/reviews_Electronics_5.json.gz
[5] https://radimrehurek.com/gensim/models/word2vec.html

Model Implementation

The programming language we used for the implementation of Aspect Extraction and Aspect Sentiment Detection models is Python 3.6 along with the deep learning framework PyTorch.[6]

The libraries we used to complete this task are:

- Ekphrasis[7]
    - Text processing tool, geared towards text from social networks. Ekphrasis performs tokenization and spelling correction based on two large corpora (English Wikipedia, twitter – 330 million English tweets).
- Numpy[8]
- Sklearn[9]
- Gensim Word2Vec

Most of the experiments ran in personal computers using GPU with CUDA. Also in order to produce our own word embeddings, as well as apply transfer learning, we used a server[10] of AUEB's NLP group.[11]

All of the code base for our models as well as Transfer Learning can be found in gitlab[12] and github[13] respectively.

3.3 Evaluation Measures and Baselines

3.3.1 Evaluation Measures

This thesis follows the guidelines of SemEval 2016 as if we were participating in the competition. The competition chose two different measures to evaluate the competency of each submission for aspect extraction and aspect sentiment detection. For aspect extraction the used

[6] https://pytorch.org/
[7] https://github.com/cbaziotis/ekphrasis
[8] http://www.numpy.org/
[9] http://scikit-learn.org/stable/index.html
[10] Server Specs CPU : Intel(R) Core(TM) i7-5930K CPU @ 3.50GHz 6-core, RAM : 32GB RAM (2 * 16Gb DIMM, 2400 MHz) DDR4, HDD : 2 x Seagate Barracuda 7200.12 3TB SATA 6Gb /s 64MB Cache , SSD : Intel 512GB, 2.5-inch SSD, M.2 SATA, GPU : 2 x ASUS NVIDIA GTX 1080 8-GB
[11] http://nlp.cs.aueb.gr/
[12] https://gitlab.com/dimitrisl/Starting_Area
[13] https://github.com/dimitrisl/TransferLearning

metric was micro-F1. In order to calculate this metric one must first get the True Positives, False Positives as well as False Negatives for the prediction of each E#A tuple. Using the previous counts we get the micro-Precision and micro-Recall which are essential in order to calculate micro-F1 since the formula is:

$$microF1 = 2 * \frac{microPrecision * microRecall}{microPrecision + microRecall}$$

In order to get micro F1 we must first calculate micro Precision and micro Recall, so for n classes we have:

$$microRecall = \frac{TP1 + \cdots + TPn}{(TP1 + \cdots + TPn) + (FN1 + \cdots + FNn)}$$

$$microPrecision = \frac{TP1 + \cdots + TPn}{(TP1 + \cdots + TPn) + (FP1 + \cdots + FPn)}$$

It is evident that micro F1 is based on global precision and recall, it treats each test case equally and does not give advantages to small classes, unlike macro-averaged F1 that could have also been used.

For Sentiment Detection the evaluation metric is Accuracy.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

3.3.2 Baselines

In order to measure the effectiveness of the implementations, SemEval has provided the contestants with baselines for all slots and all the datasets [6]. Apart from the competition's baseline, we used the scores achieved by the winners as our very own strong baselines to measure the capabilities of our implementations and methods. For aspect extraction the metric used is micro F1 and for Aspect Sentiment Detection accuracy, since those were the guidelines suggested by the competition organizers.

Aspect Detection	Micro F1
SVM (SemEval Baseline)	37.48%
NLANG CNN (SemEval Winners)	51.93%

Table 3 - Aspect Extraction Baselines.

Aspect Sentiment Detection	Accuracy
SVM classifier (SemEval Baseline)	70.03 %
SemEval Winners IIT SVM and probabilistic models	82.77 %

Table 4 - Sentiment Detection Baselines.

As already mentioned in Chapter 2, SemEval has created two baseline methods in order to help the submitting teams measure the efficiency of their models, described in the following paragraphs.

For AE, an SVM with a linear kernel is trained. In particular, *n* unigram features are extracted from the respective sentence of each tuple that is encountered in the training data. The category value (e.g., "laptop#general") of the tuple is used as the correct label of the feature vector. Similarly, for each test sentence *s*, a feature vector is built and the trained SVM is used to predict the probabilities of assigning each possible category to *s* (e.g., {"laptop#general", 0.2}, {"laptop#cpu", 0.4}). Then, a threshold *t* is used to decide which of the categories will be assigned to *s* [6]. For ASD the baseline is an SVM classifier with a linear kernel. Again, as in

AE, *n* unigram features are extracted from the respective sentence of each tuple of the training data. In addition, an integer-valued feature that indicates the category of the tuple is used. The correct label for the extracted training feature vector is the corresponding polarity value (e.g., "positive"). Then, for each tuple of a test sentence *s*, a feature vector is built and classified using the trained SVM.

Besides the baselines, we consider as strong baselines the winning systems. For AE, a feed forward network that achieved 51.93% micro F1 [15] and for ASD a system incorporating domain dependency graph[14] features [17], a distributional thesaurus and unsupervised lexical induction [2] that achieved 82.77% accuracy.

3.4 Models

3.4.1 Self Attention– Bi LSTM for Aspect Extraction

Methodology Background

Recurrent Neural Networks (RNN) are an extension of conventional feed-forward networks In this thesis, we use RNNs to model the sequence of words in a text. However standard RNNs suffer from vanishing gradient problems. To overcome these problems, the Long Short-term Memory network (LSTM) [8] was developed and achieved superior performance [8]. An LSTM is well-suited to classify and predict time series given time lags of unknown size and duration between important events.

Aspect Model (LoLSTM)

In this architecture the LSTM serves as a classifier. In order to enhance this classification process, a self-attention mechanism is applied to complement the LSTM functionality of AE. The self-attention layer is a method that enables you to "glance" back at previous states, but on a weighted combination of all the LSTM's hidden states. Figure 3 represents the architecture of a self-Attention Bi LSTM (AT-Bi LSTM).

[14] Directed graph with labeled nodes and labeled edges, constructed by aggregating individual dependency relations between domain-specific content words.

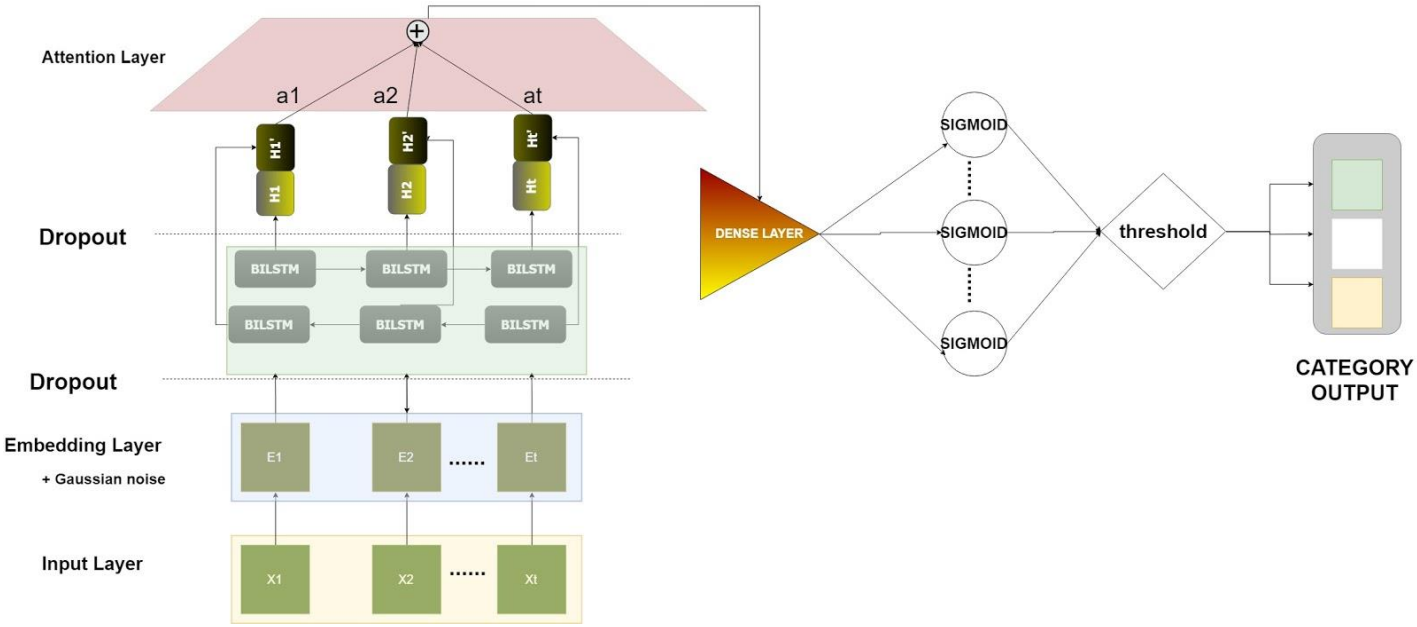

Figure 3 - LSTM with self-Attention mechanism (LoLSTM).

More formally each cell in LSTM can be computed as follows:

$$X = \begin{bmatrix} h_{t-1} \\ x_t \end{bmatrix}$$

$$f_t = \sigma(W_f * X + b_f)$$

$$i_t = \sigma(W_i * X + b_i)$$

$$o_t = \sigma(W_o * X + b_o)$$

$$c_t = \sigma(W_f * X + b_f)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tanh(W_c * X + b_c)$$

$$h_t = o_t \odot \tanh(c_t)$$

Where $W_i, W_f, W_i \in \mathbb{R}^{d*2d}$ are the weighted matrices, $b_i, b_f, b_o \in \mathbb{R}^d$ are the biases of LSTM to be learned during training, σ is the sigmoid function, $\odot$ stands for element-wise multiplication, and h_t is the hidden state of timestep t. Let $H \in \mathbb{R}^{d*N}$ be a matrix consisting of hidden vectors $[h_1, h_2, h_3, \ldots, h_N]$ that the LSTM produced, where d is the size of the hidden layers, N is the length of the given sentence and w represents the word vector.

The attention mechanism will produce an attention weight vector a and a weighted hidden representation r.

$$M = \tanh(W_h H)$$

$$a = softmax(w^T M)$$

$$r = Ha^T$$

Where $\in \mathbb{R}^{d*N}$, $a \in \mathbb{R}^N, r \in \mathbb{R}^d, W_h \in \mathbb{R}^{d*d}$.

The whole architecture consists of the input layer, the embedding layer, LSTM (bi-directorial), the attention mechanism and finally the output of this multi label classification which is a sigmoid. The input layer contains the words of each sentence; the embedding layer takes these words and projects them to 300-dimensional pre-trained embeddings. After giving the input to the bidirectional LSTM the attention layer assigns more importance (higher attention score) to the recurrent representations of the words (LSTM states) that correspond to words expressing aspects.

In our case, r has 82 dimensions, as many as the aspect categories. After getting r, we apply a sigmoid to each one of its elements, to obtain a probabilistic prediction per category. We use a threshold of 0.5 for the final decision per category, i.e., we predict that the sentence expresses an aspect if the corresponding probabilistic prediction of the model is above 0.5. We actually use mini-batches of 10 sentences.

3.4.2 LoCNN for Sentiment Polarity

Methodology Background

In machine learning, a CNN [16] is a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery [1]. CNNs were inspired by biological processes in that the connectivity pattern between neurons resembles the organization of the animal visual cortex. Individual cortical neurons respond to stimuli only in a restricted region of the visual field known as the receptive field. The receptive fields of different neurons partially overlap such that they cover the entire visual field. Recent papers indicate that CNNs are very competent also in text classification [13].

Aspect Embedding

In order to make the CNN able to understand the sentiment for each aspect, we used an Aspect Embedding. To create the aspect embedding as shown in Figure 4, we split the aspect entity tuple to its constituent words tokens (e.g. “LAPTOP#GENERAL” becomes “laptop”, “general”). Afterwards, we look up the embeddings of each word and average them to retrieve the initial value of the aspect embedding, which is further trained (via backpropagation) when training the model. For aspect-based sentiment analysis, we feed the aspect vector together with the word embeddings of the input sentence into a CNN. The intuition behind this mechanism is that the model will be guided to correlate the aspect of a sentence with its sentiment.

Sentiment Model (LoCNN)

This model architecture is an extension of INSIGHT [9]. The whole architecture is depicted in Figures 4, 5 and 6. The CNN takes a text as input, which is padded to the maximum sentence length *m* of each batch. The text is represented as a concatenation of its word embeddings $x_{1:m}$ where $x_i \in R^k$ is the *k*-dimensional vector of the i^{th} word in the text. After going through the embedding layer, the convolution layer slides kernels of different sizes over the input embeddings. Each filter creates a new feature (c_i) for a window of *h* words; the application of the filter over each possible window of *h* words in the sentence produces a feature map.

$$c_i = f(w * x_{i:i+h-1} + b)$$

Note that $b \in \mathbb{R}$ is a bias term and *f* is a nonlinear function, ReLU.

$$c = [c_1, c_2, c_3, \dots, c_{n-h+1}]$$

Max-over time pooling in turn condenses this feature vector to its most important feature by taking its maximum value and dealing with variable input lengths. The loss function we use for the model training is Cross Entropy. A final softmax layer takes the concatenation of the maximum values of the feature maps produced by all filters and outputs a probability distribution over the three (positive, neutral, negative) output classes.

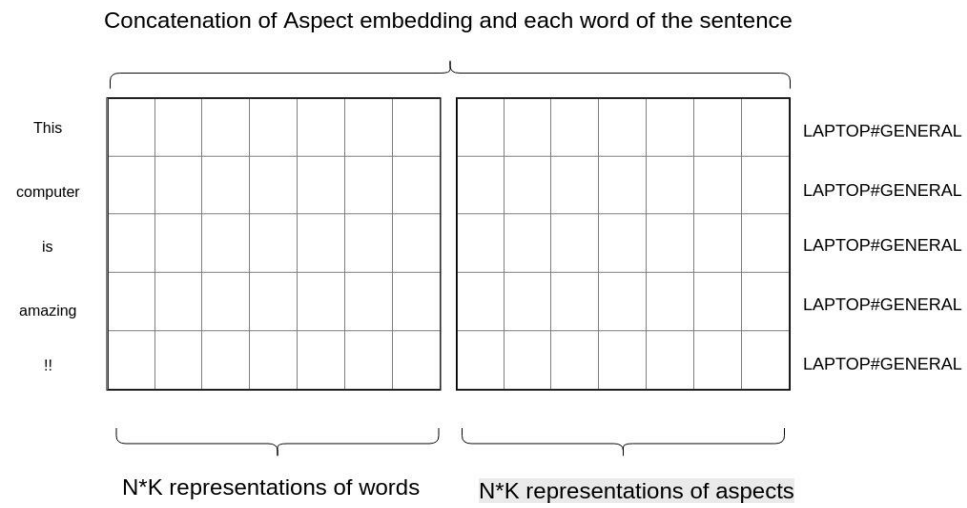

Figure 4 - Aspect Embedding.

The original Aspect Embedding is represented in a 300 dimensional space, but not all of these dimensions are useful. In order to make the aspect embedding vector more relevant we decided to use a Linear Transformation to project the 300 dimensions to 15 as shown in Figure 5.

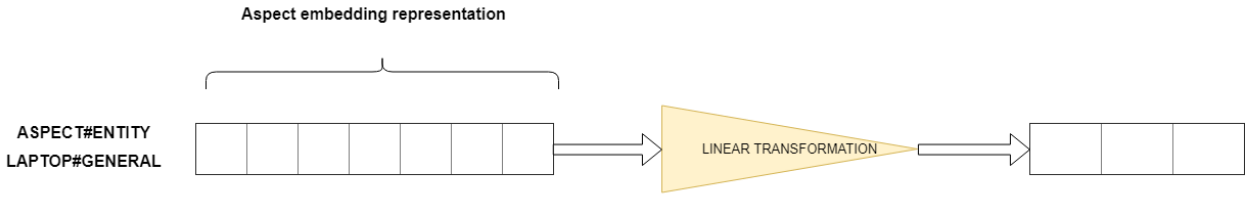

Figure 5 - Size reduction.

Overall, the whole architecture is depicted in Figure 6.

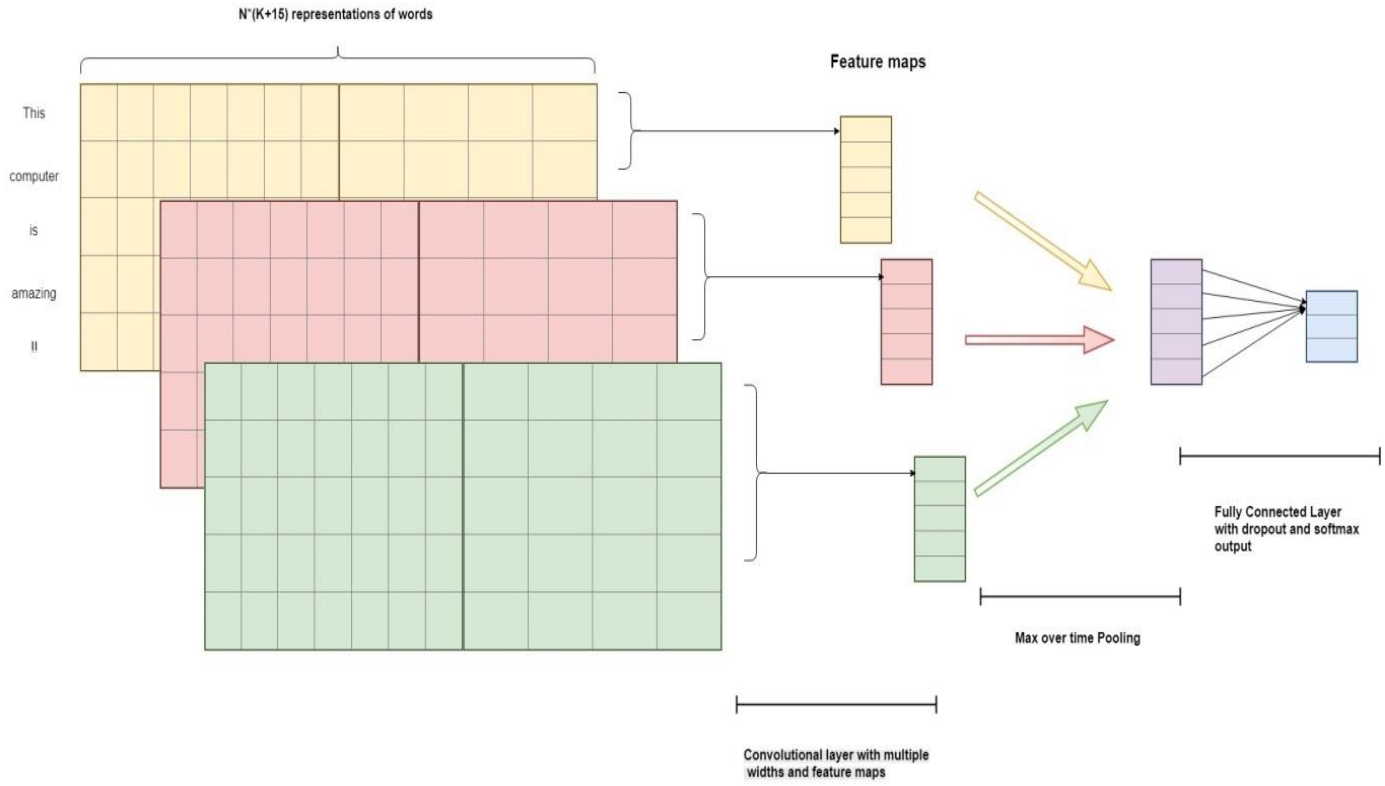

Figure 6 - CNN with Aspect Embedding (LoCNN).

Even though the architecture of Figure 6 provided us with results that were comparable to the results of the INSIGHT team [9], in order to surpass their submission we decided to use Transfer Learning. The results of this method improved the performance of our model, therefore, all the results of the finally reported experiments were achieved using transfer learning.

Transfer Learning

Transfer learning (Figure 7) is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. As already mentioned before one of the challenges we faced was the small dataset and even though the data augmentation helped the validation results significantly (see Section 3.2), the LoCNN still couldn't achieve accuracy scores comparable to those reported in [9]. Transfer learning was the ideal option since we possessed a large annotated electronics reviews dataset from Amazon which was conceptually closer to the training dataset. The Amazon dataset did not contain any of our original aspects or even the tags that we used for sentiment polarity. In order to tackle that issue we replaced the

aspects of the reviews with the aspect “ALL” and used as its Aspect Embedding the average of all aspect embeddings of our original schema. We also changed the 5 scale rating of these reviews to negative (reviews containing less than 3 stars), neutral (reviews containing 3 stars) and positive (reviews that contain more than 3 stars). The intuition behind this substitution is that since the Amazon dataset did not contain the same Aspect Entity tuples with the SemEval dataset, the “ALL” aspect would influence our model towards the original aspects. After reaching the epoch that achieves the best results on the Amazon dataset, we use the weights to initialize LoCNN which is then trains these weights on the original dataset through back propagation.

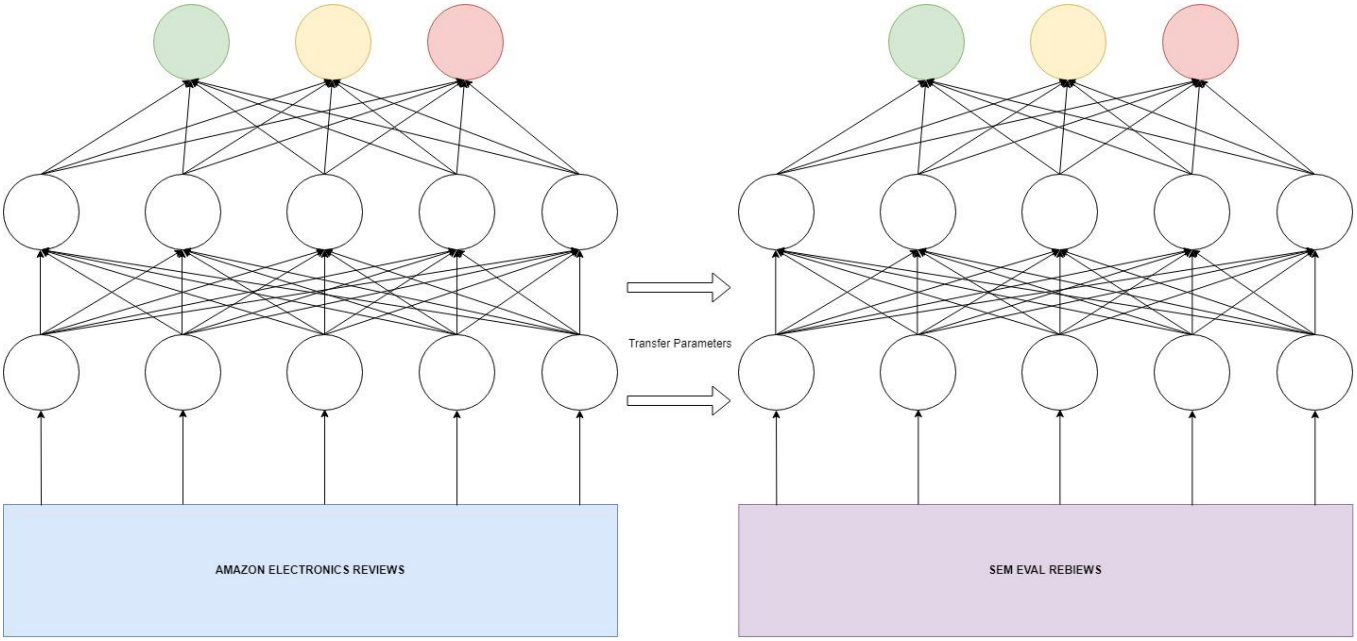

Figure 7 - Transfer Learning.

Chapter 4 – Experiments

4.1 Aspect Extraction

Using the architecture of Section 3.4.1 which was inspired by the works of [14] and [5], we began training the model using the initial dataset provided by SemEval 2016 on the laptops dataset. This dataset contained 2000 opinionated sentences out of which we used 70% for training and 30% for validation. The model was trained for several epochs until it was starting to overfit; to get a better vision of the model's behavior we trained the model for 80 epochs even though its peak performance happened before we reached the 30th epoch for the SemEval dataset (Figure 8). For the Datasets SemEval#GE & SemEval#GE+FR+SP the model started to overfit after the epochs 49 and 71 respectively (Figures 12 and 14).

Finally training the model with SemEval#GE+FR+SP (Figure 14), we managed to surpass both the baseline and the winning submission of the competition with a micro F1 score of 53.50% at epoch 71 where the validation scores are the highest.

Experiment procedure

In this section we report the models' performance on the development set of SemEval Dataset. In Table 5 we show the values of the hyper-parameters we have used, that came out of manual tuning in the training dataset.

Hyper Parameters	
RNN size	150
RNN layers	1
Gaussian Noise[15]	0.5
Dropout words[16]	0.5
Dropout RNN	0.5
Batch Size	10

Table 5 - LoLSTM Hyperparameters.

[15] Statistical noise applied to the word embeddings with $\mu = 0$ and σ the value mentioned in the hyperparameters.
[16] The dropout layer of the word embeddings.

Dataset SemEval

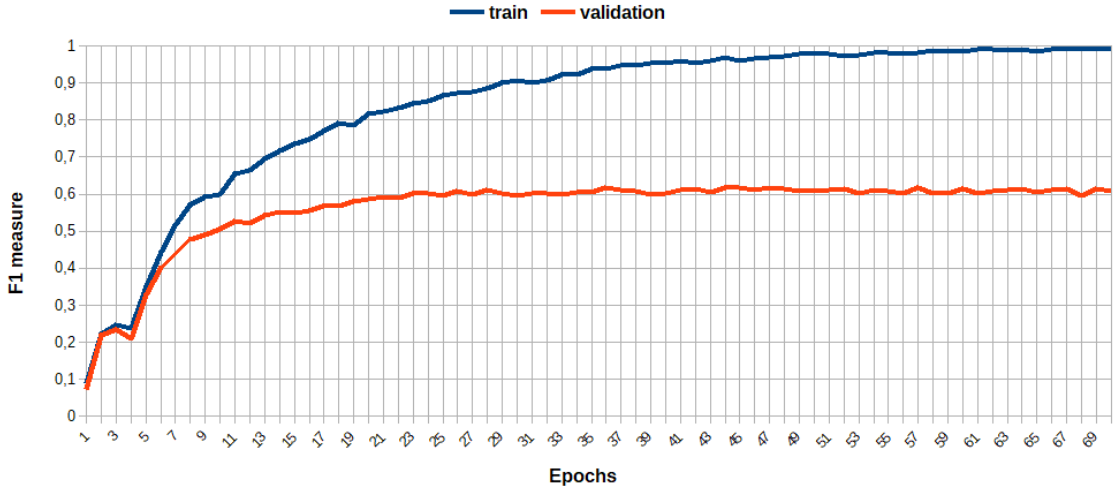

Figure 8 - Micro F1 of LoLSTM on the SemEval dataset.

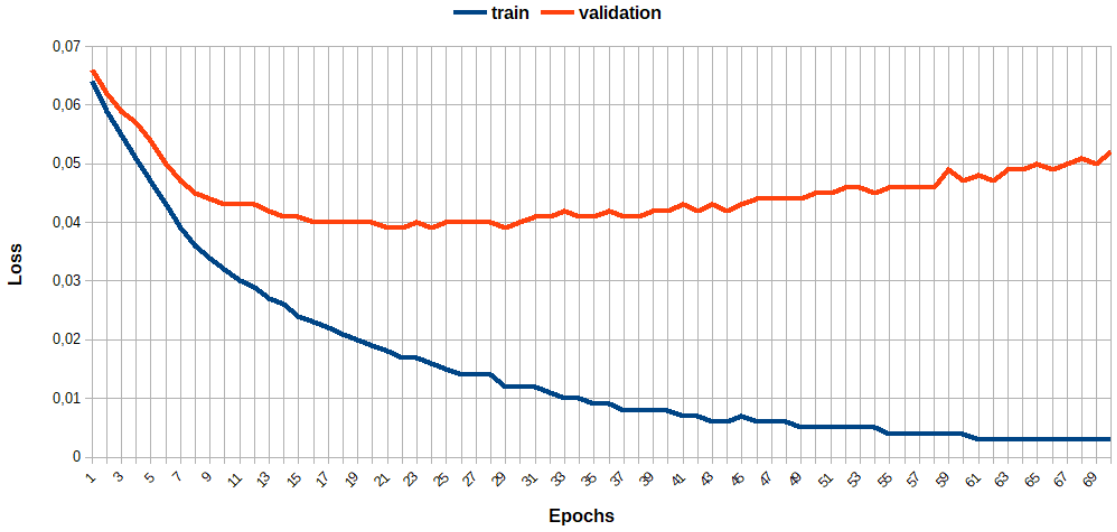

Figure 9 - LoLSTM Loss for SemEval.

As we can see in Figure 9, the model starts to overfit right after the 19th epoch. After that epoch micro F1 does not seem to rise in the validation set, which is one of the signs of overfitting,

whereas the micro F1 continues to improve on the training data, which is also a sign of overfitting.

4.2 Sentiment Polarity

Experiment Procedure

Following the same procedure as with AE, the hyper parameters were tuned on the training dataset and are shown in Table 6.

Hyper Parameters	
Kernel dimension	30
Kernel sizes	3,4,5
Gaussian Noise	0
Dropout words	0.2
Batch Size	15

Table 6 - LoCNN hyperparameters.

Dataset SemEval

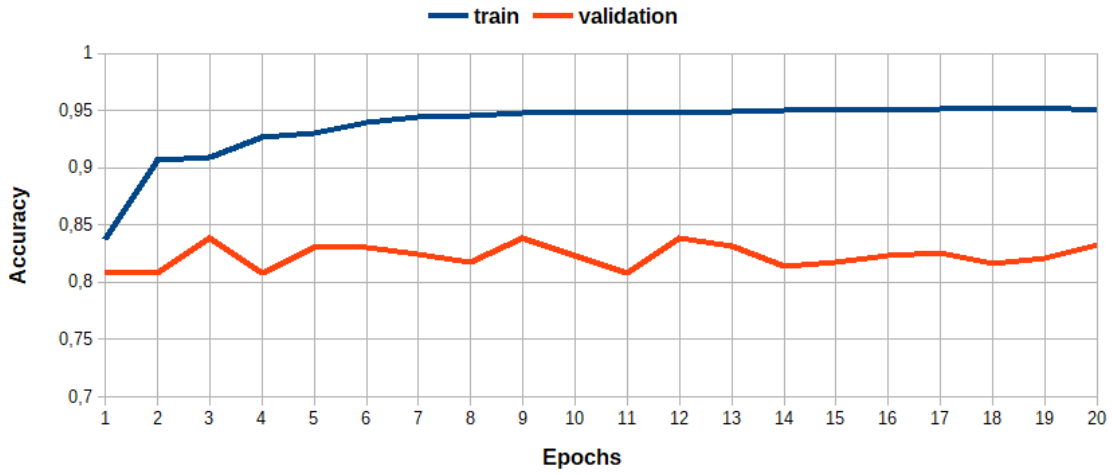

Figure 10 – Accuracy metric LoCNN on the SemEval dataset.

Using the original dataset provided by SemEval we achieve 83% accuracy in the 12th epoch. As with the rest of the experiments we stopped training the LoCNN at the 20th epoch.

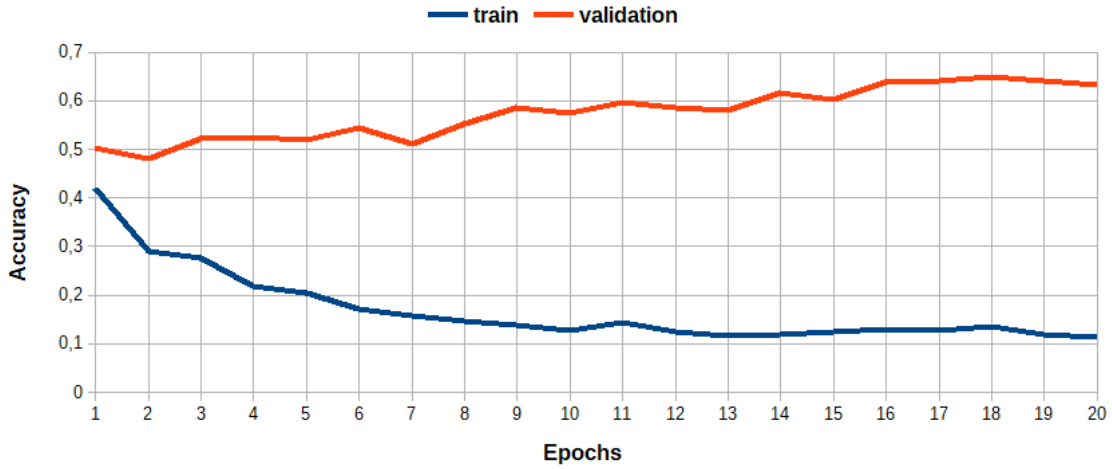

Figure 11 - LoCNN Loss on the SemEval dataset.

4.3 Data Augmentation

We used the original dataset, and we translated it using Google API,[17] to French, German and Spanish and back again to English, in order to augment it. This method is an easy way of enlarging the training set without having to annotate more raw data. The intuition behind this data augmentation is that the pivoted data will be similar but not the same with the original text, thus increasing the size of the dataset. With data augmentation the aspect ratios as well as sentiment ratios in the enlarged datasets remain the same (see Figures 1 and 2). The dataset enlargement was applied in AE as well as in the ASD, with great results (see Sections 4.3.1 and 4.3.2).

[17] https://cloud.google.com/translate/docs/apis

4.3.1 AE with Augmented Datasets

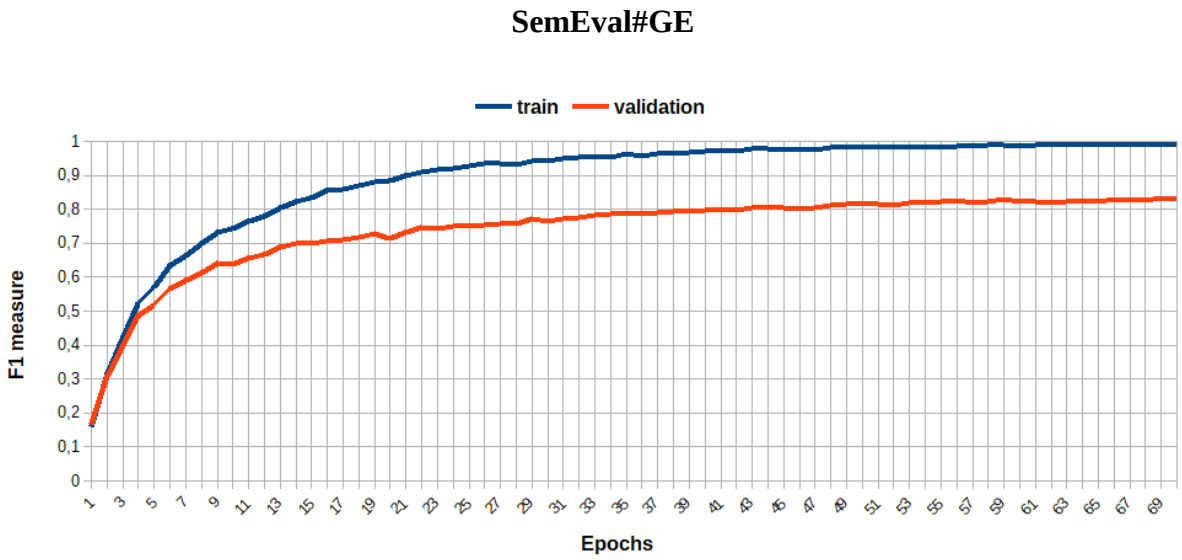

Figure 12 - F1 metric LoLSTM for SemEval#GE.

Comparing Figures 8 and 12, the data augmentation provides a significant rise in both train and validation scores. More specifically in Figure 8 the highest F1 score achieved was 60% while in Figure 12 the highest F1 score was 80%.

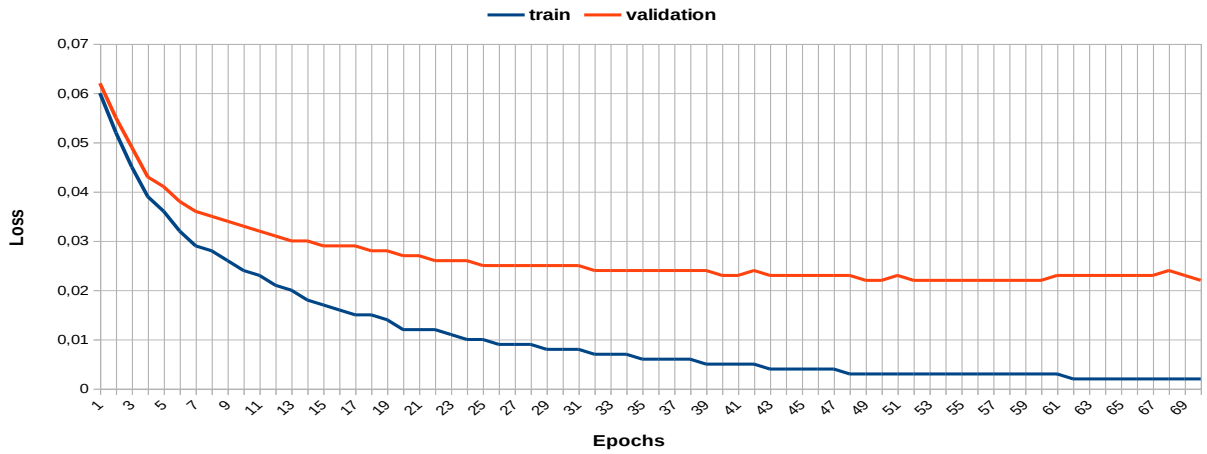

Figure 13 - LoLSTM Loss for SemEval#GE.

SemEval#GE+FR+SP

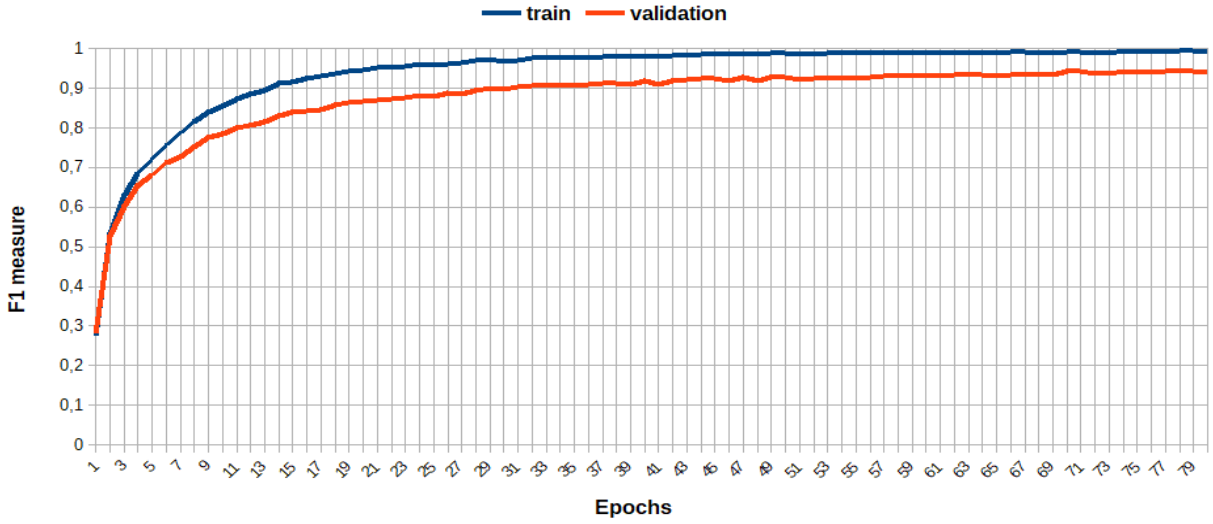

Figure 14 - F1 metric LoLSTM for SemEval#GE+FR+SP.

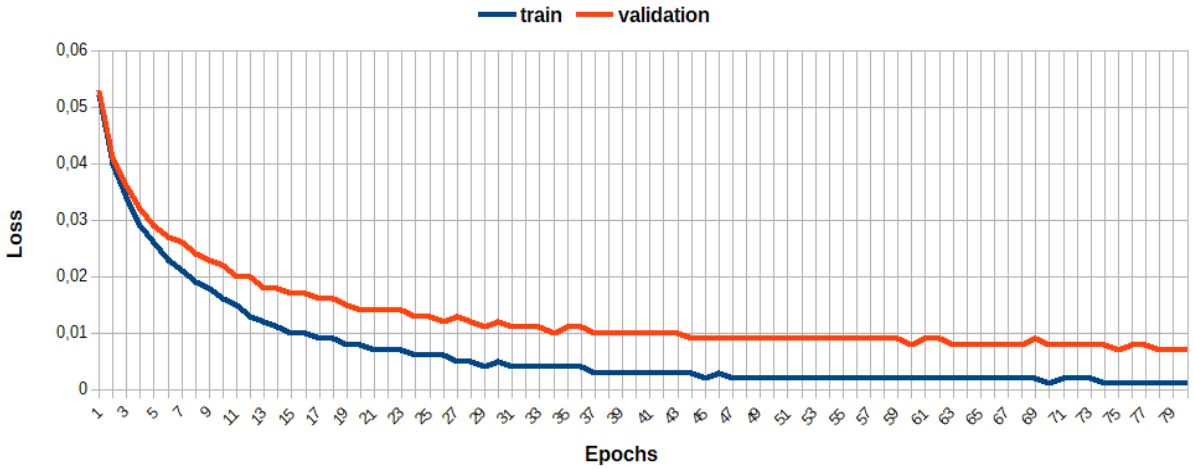

Figure 15 - LoLSTM Loss for SemEval#GE+FR+SP.

There is apparent difference with the triple size of the original dataset when comparing the results shown in Figures 8, 12 and 14. Overfitting is reduced further, again due to the increase in the size of the training set, and the best micro F1 score on the development set increases from 61% in Figure 8 and 81% in Figure 12 to 93% in Figure 14.

In these tests the model achieved a micro F1 score of 53.50% which is 1.63% more than the winners of 2016 SemEval's winners [15] who achieved micro F1 of 51.93%. All in all, the dataset expansion seemed to have a positive effect in getting better results.

4.3.2 ASD with Augmented Datasets

In ASD, doubling the dataset again seems to produce significantly better results than those obtained with the original dataset. The accuracy at the best epoch (14th epoch) is now 87% (Figure 16), a rise of 4% compared to the best accuracy of Figure 10.

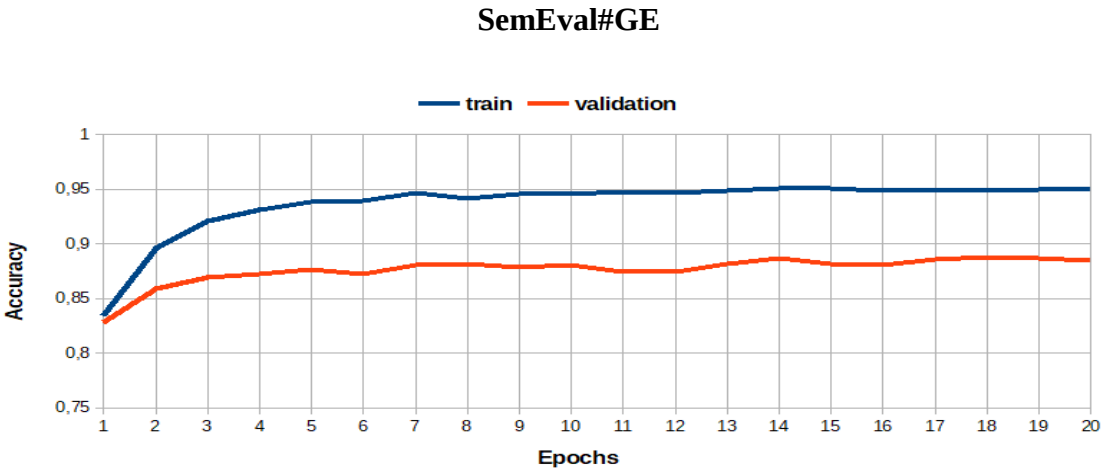

Figure 16 - Accuracy metric LoCNN on SemEval#GE.

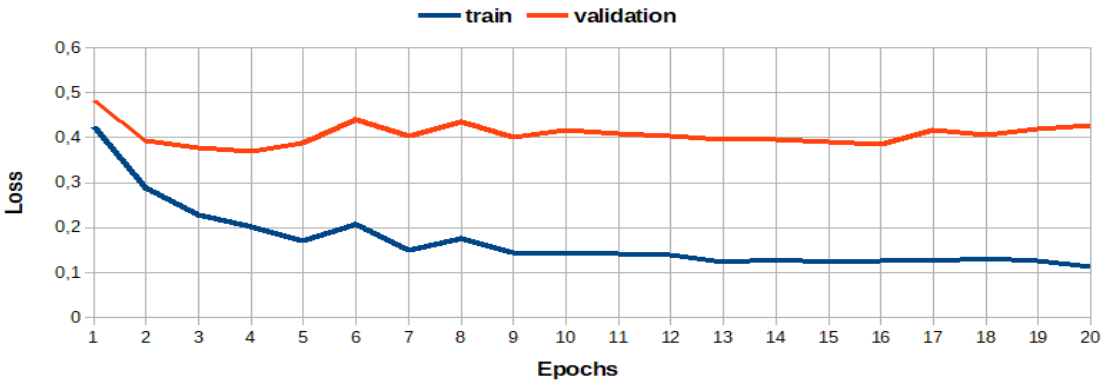

Figure 17 - LoCNN Loss on SemEval#GE.

Further augmentation of the dataset with French does not seem to produce significantly higher results than English and German (SemEval#GE), it does seem though to make the behavior of the system more stable (smother curves). The 13th epoch of this dataset (Figure 18) still produces a higher score than the 14th epoch of the previous dataset.

SemEval#GE+FR

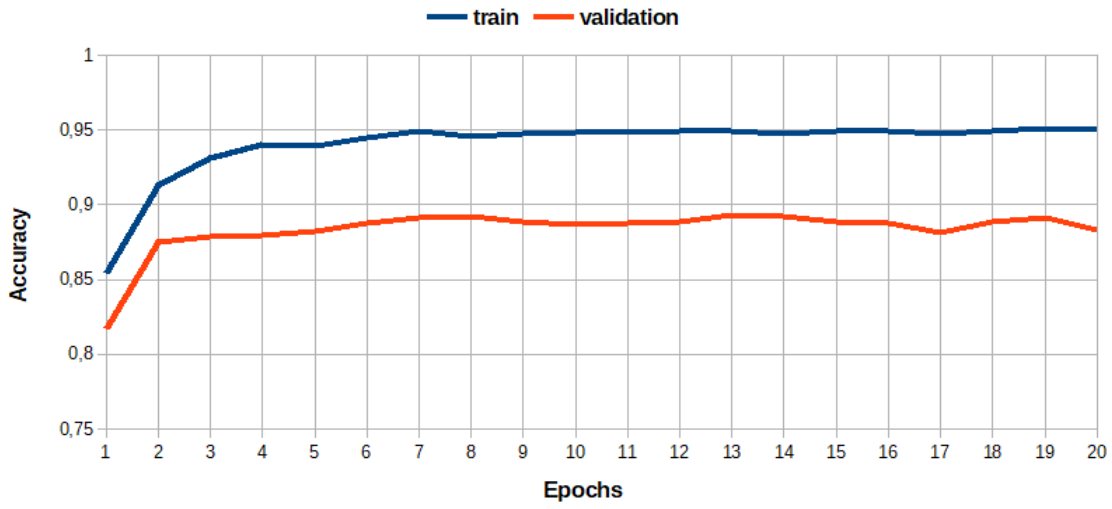

Figure 18 - Accuracy metric LoCNN on SemEval#GE+FR.

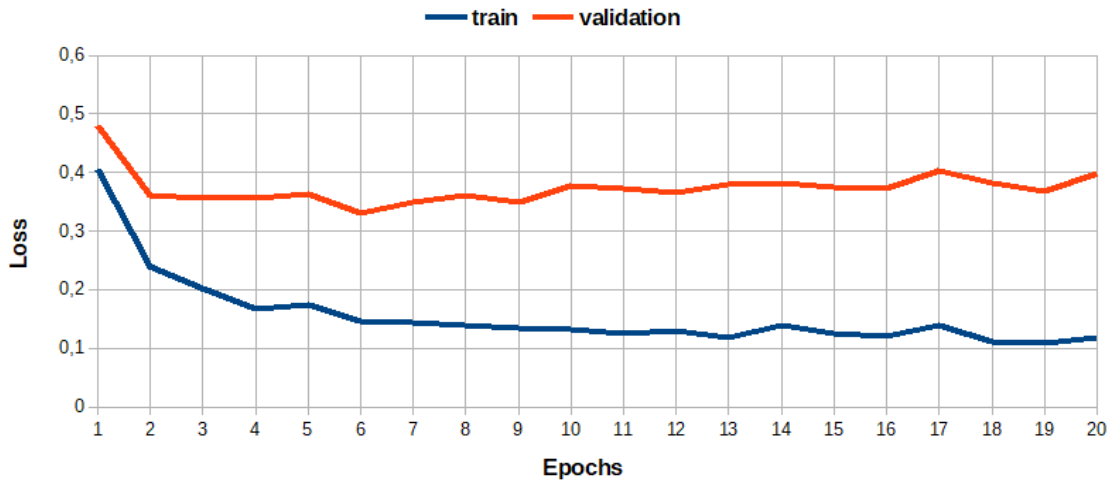

Figure 19 - LoCNN Loss on SemEval#GE+FR.

As it is apparent from Figure 20, the more data we feed the model the better results it produces. This particular dataset is 4 times larger than the original and it has provided us with 90% accuracy in the validation set. Compared to the original dataset we have a 7% rise when we compare the best epochs of each.

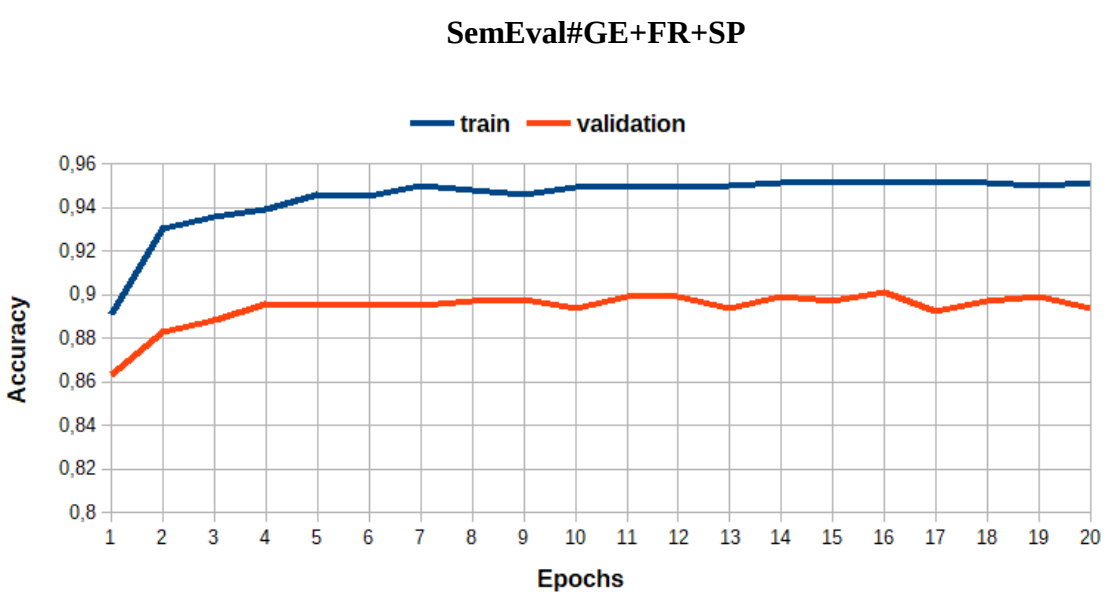

Figure 20 - Accuracy metric LoCNN on SemEval#GE+FR+SP.

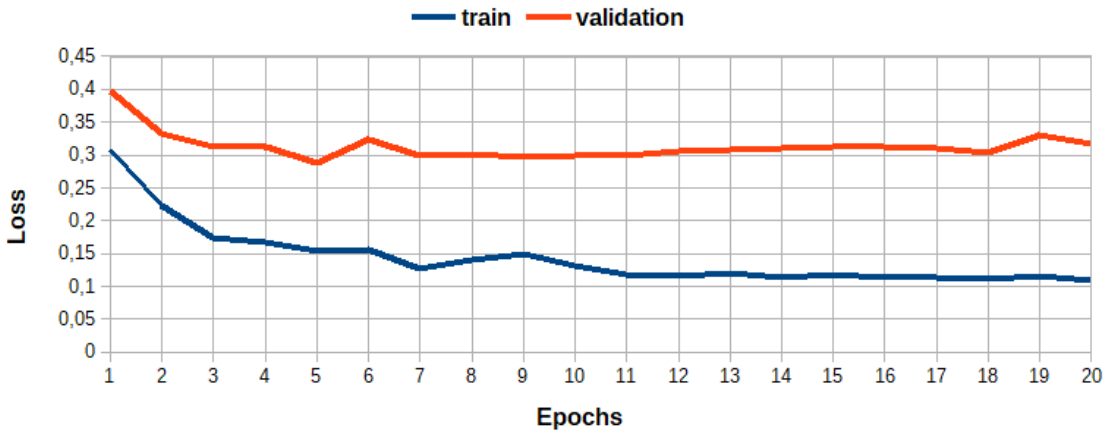

Figure 21 - LoCNN Loss on SemEval#GE+FR+SP.

After the experimentation in all those datasets, we can conclude that that data augmentation by translating to pivot languages and back has the potential to lead to very significant performance improvements, at least in the tasks we have considered.

Finally, using the dataset that performed the best out of the 4 (the SemEval#GE+FR+SP as train dataset), we achieved the score of 79.65% accuracy which is higher than the one achieved by the team we replicated and modified methods from [9]. Their model achieved 78.40%. Even though this score would rank us second in the competition, it could not achieve first place (82.77%).

Chapter 5 – Conclusions and Future Work

5.1 Conclusions

The systems proposed in this thesis achieve great results in AE as well as ASD by using approaches that to our knowledge have not been tested in the SemEval 2016 competition. Especially in the AE task we surpass the winning submission by 1.63% of micro F1. The ASD task proved to be harder to overcome without using any feature engineering, like the winners. Nevertheless, we achieved 79.65% accuracy score, which would place us in the 2nd position in the corresponding task of SemEval 2016. We also note that we have successfully applied ABSA with deep learning techniques to small datasets using data augmentation as well as transfer learning.

During the testing phase of this thesis, apart from the main structure of the models we decided to try several methods in order to raise the efficiency and coverage of our systems. For AE, data augmentation led to very significant improvements, allowing us to surpass the results of the best SemEval system. On the other hand ASD, due to the existence of conflicting sentiments for different aspects in each sentence, made it difficult to perform better than the winning submission from SemEval, and as a result we tried to use Deep Residual Learning [18] and Part of Speech Embeddings [19], without success.

5.2 Future Work

As we have mentioned in Chapter 2 a lot of research has been done in the sentiment analysis area, and a lot of those methods were tried in the SemEval 2016 competition. Taking that into account we can only offer some suggestions for further research and improvements to our implementations since time did not permit to apply them. We first list the suggestions and then explain them further.

For Aspect Extraction:

- Transfer Learning
    - Already used in Sentiment Detection, transfer learning could be helpful in Aspect Extraction too.
- Ensemble Voting.
- Handcrafted features.
    - Distributional Thesaurus.

For Sentiment Detection:

- Ensemble Voting.
- A LSTM implementation with an attention layer.
- Handcrafted features.
    - Distributional thesaurus.
    - Emotional Lexicons.

Ensemble methods are techniques that create multiple models and then combine them to produce improved results. The ensemble vote classifier as depicted in Figure 22 is a classifier that usually combines different machine learning models for classification tasks. In this case we could actually run the same models with different seeds and different starting weights, in order to obtain different models. The Ensemble Vote Classifier implements "hard" and "soft" voting. In hard voting, it predicts the final class label as the class label that has been predicted most frequently by the classification models. In soft voting, it predicts the class labels by averaging the class-probabilities.

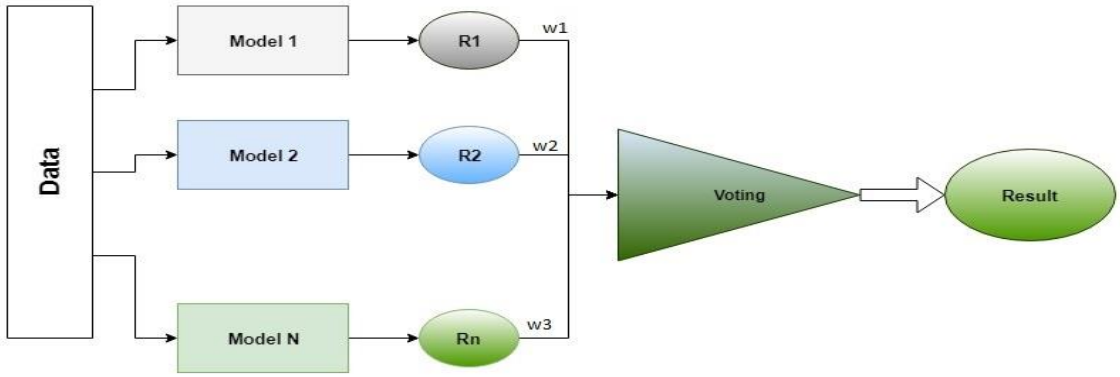

Figure 22 – Voting.

For Sentiment Detection an LSTM with an attention layer could potentially have interesting results. To be more precise, an architecture like the one in Figure 3 could be transformed to be applied in Sentiment Detection too, with the addition of the aspect embedding in the input and hidden state.

A Distributional Thesaurus (DT) is an automatically computed lexical resource that ranks words according to their semantic similarity to other known useful words. For every top five significant words (based on tf idf score) in each aspect category (for example: 'overpriced', '$', 'pricey', 'cheap', 'expensive' are the most significant terms in 'laptop#price' category), we find the ten most similar words according to DT. The presence or absence of these words in the review is used as a feature for aspect category identification [2].

Finally, ASD emotional Lexicons work by constructing a polarity lexicon from large external corpora related to the topic at hand and promotes words depending on their polarity.

Bibliography

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," *Adv. Neural Inf. Process. Syst.*, pp. 1–9, 2012.

[2] A. Kumar, S. Kohail, A. Kumar, A. Ekbal, and C. Biemann, "IIT-TUDA at SemEval-2016 Task 5: Beyond Sentiment Lexicon: Combining Domain Dependency and Distributional Semantics Features for Aspect Based Sentiment Analysis," *SemEval*, pp. 1129–1135, 2016.

[3] A. Severyn and A. Moschitti, "UNITN: Training Deep Convolutional Neural Network for Twitter Sentiment Classification," *SemEval 2015*, pp. 464–469, 2015.

[4] D. Xenos, P. Theodorakakos, and J. Pavlopoulos, "AUEB-ABSA Ensembles of Classifiers and Embeddings for Aspect Based Sentiment Analysis", *SemEval-2016 Task 5*, pp. 312–317, 2016.

[5] B. Wang and M. Liu. *Deep Learning for Aspect-Based Sentiment Analysis*, report, Stanford University, 2015.

[6] M. Pontiki, D. Galanis, H. Papageorgiou, I. Androutsopoulos, S. Manandhar, M. AL-Smadi, M. Al-Ayyoub, *"*SemEval-2016 Task 5 : Aspect Based Sentiment Analysis.*" In Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval-2016),*San Diego, California, pp. 19–30 , *2016.*

[7] R. Collobert and J. Weston, "A unified architecture for natural language processing," in *Proceedings of the 25th International Conference on Machine learning - ICML '08*, pp. 160–167, 2008.

[8] S. Hochreiter and J. Urgen Schmidhuber, "long short term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] S. Ruder, P. Ghaffari, and J. G. Breslin, "INSIGHT-1 Deep Learning for Multilingual Aspect-based Sentiment Analysis,", at *SemEval-2016*, pp. 330–336, 2016.

[10] T. Mikolov, G. Corrado, K. Chen, and J. Dean, "Efficient Estimation of Word Representations in Vector Space," *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, pp. 1–

12, 2013.

[11] T. H. Nguyen and K. Shirai, “PhraseRNN: Phrase Recursive Neural Network for Aspect-based Sentiment Analysis,” *Proc. 2015 Conf. Empir. Methods Nat. Lang. Process.*, pp. 2509–2514, 2015.

[12] T. Khalil and Samhaa R. El-Beltagy, “NileTMRG: Deep Convolutional Neural Networks for Aspect Category and Sentiment Extraction in SemEval-2016 Task 5,” *Proc. 10th Int. Work. Semant. Eval. (SemEval 2016)*, 2016, pp. 276–281, 2016.

[13] Y. Kim, “Convolutional Neural Networks for Sentence Classification” , Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, Doha, Qatar, pp.1746-1751, 2014.

[14] Y. Wang, M. Huang, L. Zhao, and X. Zhu, “Attention-based LSTM for Aspect-level Sentiment Classification,” in *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing (EMNLP*, pp. 606–615*)*, 2016.

[15] Z. Toh and J. Su, “NLANGP at SemEval-2016 Task 5: Improving Aspect Based Sentiment Analysis using Neural Network Features,” *Proc. SemEval-2016*, vol. 2015, no. Subtask 1, pp. 282–288, 2016.

[16] Y. LeCun, P. Haffner, L. Bottou, Y. Bengio “Object Recognition with Gradient-Based Learning. In: Shape, Contour and Grouping in Computer Vision”. *Lecture Notes in Computer Science*, vol 1681. Springer, Heidelberg, 1999.

[17] S. Kohail and T. U. Darmstadt, “Unsupervised Topic-Specific Domain Dependency Graphs for Aspect Identification in Sentiment Analysis”, *International Conference Recent Advances in Natural Language Processing, RANLP*, pp. 16–23, 2015.

[18] S. Wu, S. Zhong, and Y. Liu, “Deep residual learning for image steganalysis,” *Multimedia Tools and Applications*, pp. 1–17, 2017.

[19] M. Miwa and M. Bansal, “End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures”, *CoRR*, abs/1601.00770, *2016.*