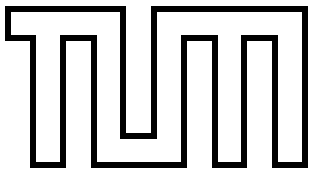


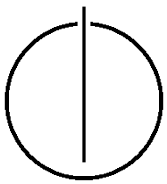
DEPARTMENT OF INFORMATICS

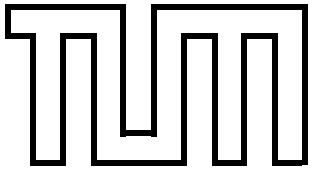
TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Deep Learning Approaches for Opinion Mining on Conversational Social Media Texts

Sudeshna Dasgupta





DEPARTMENT OF INFORMATICS

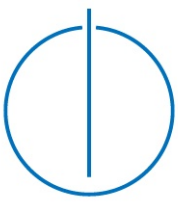
TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Informatics

Deep Learning Approaches for Opinion Mining on Conversational Social Media Texts

Ansätze des tiefen Lernens für Opinion Mining auf Texten aus dialogorientierten sozialen Medien

Author:	Sudeshna Dasgupta
Supervisor:	Prof. Dr. Georg Groh
Advisor:	Gerhard Hagerer, M.Sc.
Submission Date:	October 15, 2020



I confirm that this Master's Thesis is my own work and I have documented all sources and material used.

Munich, October 15, 2020 Sudeshna Dasgupta

Acknowledgement

This thesis titled "Deep Learning Approaches for Opinion Mining on Conversational Social Media Texts" was completed in the year 2019/2020 at the Technical University of Munich.

First and foremost I would like to thank my supervisor Prof. Dr. Georg Groh and my advisor Gerhard Hagerer from the Research Group of Social Computing, for giving me the opportunity to write my Master's Thesis with their research group. This has been a great learning experience for me. I am very grateful to them for their guidance and support. Their knowledge, enthusiasm, and motivation have guided me and enriched me all through my research.

I especially wish to express my sincere gratitude to Gerhard Hagerer for his insightful feedback, continuous supervision, and valuable guidance. His appreciation, encouragement, and trust have helped me immensely to accomplish this thesis.

This thesis was pursued during an ongoing pandemic which forced all work to be completed remotely. I am deeply indebted to Gerhard Hagerer for being accessible for all my doubts and questions and for reaching out in unprecedented situations. His empathy, positivity and enthusiasm kept me motivated and hopeful even in difficult circumstances.

Finally, I would like to extend my gratitude and regards to my Baba and Maa for their love, care, and support. I would also like to thank my friends who have always believed in me and have been there for me.

Abstract

Opinions and reviews help individuals to make decisions. Opinion mining on social media conversational text such as forum discussions, product reviews provides valuable insight on user sentiment. In a span of review text, opinions are expressed towards aspect terms. To understand the fine-grained sentiment expressed toward each aspect in a product review, aspect term identification and aspect term categorization is needed. Jointly they are referred to as aspect extraction. The task requires datasets with annotations that are specific to the domain, and at a fine grained level. Most datasets are human annotated due to which dataset construction is manual and tedious. The task of aspect extraction has been majorly approached using machine learning methods that use hand-crafted features, but they have a drawback as rule based mining is difficult to scale over different datasets and domains. Deep learning methods have also been proposed for the task of aspect extraction, but they require a large amount of data annotated with aspect information. In this thesis, we perform an empirical study of three unsupervised deep learning models to perform aspect extraction on the Sentihood dataset. To evaluate the models we use multiple word embeddings to conduct extensive experiments. We evaluate our observations using quantitative and qualitative analysis of extracted aspects, and the micro F1 scores for the task of aspect classification. We discuss the limitations faced in our experiments, and use our findings to conclude about the methodology applied in this thesis.

Contents

1 Introduction

1.1 Motivation

Social media platforms are a raging source of opinionated text. People express their views about products, services and experiences to influence the decision of their audience. The platforms regularly receive an overwhelming amount of data from their users that has made the research field of opinion mining in social media extremely lucrative. Opinions, feedback, and critiques reflect the attitude and sentiments of humans towards a topic or service. (Wang et al., 2016) Opinions on social media are typically available as comments or reviews. Reviews are short reports about a product or a service written by a customer or user on a website, to help people decide if they want to purchase the product or the service. Companies can harness this insight to gauge ongoing trends and business opportunities. A sentiment analysis(Pang and Lee, 2008) model can analyze a review to extract the polarity of the sentiment associated with the text. Still, a review may include opinions about the specific aspects of the product such as color, quality, or price. Subsequently, an opinionated review might convey opposing sentiments, and analysis of the overall polarity of the text might not be useful. A more fine-grained approach like aspect-based sentiment analysis (ABSA) (Pontiki et al., 2014) identifies the individual sentiments expressed towards the aspects of the product being reviewed.

Aspect Extraction
Aspect extraction is a key task inclusive in the framework of ABSA. It consists of two subtasks, aspect term extraction, and aspect term clustering. In a given span of text, aspect term extraction (ATE) is the task of extracting words or expressions that explicitly represent an entity. For example, in the sentence, *'The city is very safe'*, the word *'city'* should be extracted as aspect term. The word *'safe'* can be associated to the aspect *'safety'*. In a collection of sentences or documents, aspect terms that associate to common topics are clustered to identify the aspects associated to the corpus. Jointly these two subtasks form the task of aspect extraction. Aspect extraction has been widely approached using rule-based and learning-based techniques.(Dai and Song, 2019) Rule-based approaches heavily depend on manually designed rules, and approaches that use supervised learning depend on the availability of datasets.

Most research methods on aspect extraction have been evaluated using product review datasets like the restaurant review dataset by (Ganu et al., 2009), and the laptop reviews dataset by (Pontiki et al., 2015). A major difference between a typical review text and conversational text is the presence of explicitly mentioned aspect terms. Product review texts typically consist of an explicit aspect term for which the opinion is expressed. For example:

'This pizza tastes good.'

The sentence has an explicitly mentioned aspect term *'pizza'* with an associated positive opinion. The aspect term can be clustered into an aspect category *'food'*. But for a review such as:

'The cellphone fits in my hand'

in the given example the model may not successfully extract the aspect term 'fits' and cluster it to the aspect category 'size'.

As aspect level annotation is a manual task, datasets are limited, which provides a strong motivation to adopt unsupervised approaches. Deep learning models have achieved exceptional results in sentiment analysis(Kim, 2014). (Wang and Liu, 2015) demonstrated the successful application of deep learning on ABSA. Unlike rule-based methods, deep learning models do not require manually designed features and are a feasible approach to scale over multiple domains and datasets. (He et al., 2017) have proposed the Attention Based Aspect Extraction (ABAE) model which is an unsupervised neural model for the task of aspect extraction. (Luo et al., 2019) proposed the Aspect Extraction with Sememe Attentions (AE-SA) model and the Aspect Extraction via Context-enhanced Sememe Attentions (AE-CSA) model that address the task of aspect extraction in an unsupervised environment. The models consist of a multi-sense and multi-sememe based hierarchical attention mechanism, that exploits the principle that a word can have several meanings depending on the context of the sentence or document. The AE-SA and AE-CSA models attempt to overcome certain limitations faced with ABAE, which were in alignment with our findings with the ABAE model and the Sentihood dataset.

1.2 Objective

In this thesis, our research objective to solve the task of aspect extraction on the Sentihood dataset published by (Saeidi et al., 2016). The dataset was sourced from Yahoo! Answers, a question-answering platform, and each data sample is a user

review about the local neighborhoods in London. The Sentihood dataset was the perfect match for our research question because:

- The state of the art aspect extraction methods have been evaluated majorly on product review datasets. As Sentihood is also a review dataset, the input samples emulate a common underlying pattern of product and description that is conventionally seen in review sentences. This helps to keep our research comparable to the state of the art aspect extraction models.

- Unlike the product review datasets where the reviewer usually mentions the aspect term, the Sentihood dataset is a service or location review. So users are free to express without a definite opinion target expression. This makes Sentihood similar to general conversational social media.

- The Sentihood dataset has 12 aspect labels inclusive of explicit and implicit aspects, which makes it a challenging usecase for aspect extraction.

This thesis is an empirical study of three unsupervised deep learning models that have achieved state of the art results in the task of aspect extraction on product review datasets. The objective of this thesis is to analyze and infer if the best performing models can attain comparable results on a dataset with a higher number of aspects. Deep learning models generally require a large corpus of data to learn input features. As the Sentihood dataset is relatively small for a deep learning architecture, we extend our experiments by applying transfer learning to the existing models. Transfer learning is a method used to improve task performance when there is an inadequacy of required training data. (Zoph et al., 2016)

1.3 Outline

In this section we define the overview of each chapter.

Chapter 2
In this chapter we discuss the recent research approaches applied to solve the task of aspect extraction. Here we try to portray our literature research and reason the selection of the approaches used in our thesis.

Chapter 3
Here we briefly describe the necessary background knowledge required to understand the thesis.

Chapter 4
In this chapter we provide a list of available datasets for aspect extraction. We also demonstrate that our thesis findings agree with the limitations faced in constructing the datasets.

Chapter 5
Here we describe the model architectures used in this thesis, the data preprocessing steps and the implementation details used to perform our experiments.

Chapter 6
This chapter describes the experiments performed in this thesis.

Chapter 7
In this chapter, we discuss the experiment results, our remarks for each experiment, and discuss our insights.

Chapter 8
This chapter is a summary of our findings in this thesis.

2 Related Work

We performed literature research to understand the approaches researchers have used to solve the problem of aspect extraction in an unsupervised learning environment. As our dataset is sourced from online reviews in English, and hence is related to conversational text in social media, we filtered our literature research with the key terms :
'unsupervised learning', 'aspect extraction', 'social media', 'conversation text', 'English', 'short text'

(Dai and Song, 2019) used manually designed rules based on the result of dependency parsing to extract aspect terms and opinion terms from product reviews. The extracted data is used as weak supervision to train a neural model that also trains on manually annotated ground truth data to perform aspect and opinion term extraction.

(Ma et al., 2019) address the task of aspect term extraction with a sequence-to-sequence learning framework. The words are labeled using the BIO(Begin, Inside, Outside) tagging scheme. (Da'u and Salim, 2019) also used a CNN based architecture to perform sequence labeling.

(Jebbara and Cimiano, 2019) address the problem of inadequate annotated data by proposing a zero-shot cross-lingual approach for extracting opinion target expressions. The model can be trained on data of source language and perform prediction on target language without using any labeled data.

(Wang and Pan, 2018) have proposed a transition-based adversarial network model that performs cross-lingual aspect extraction.

(López and Arco, 2019) use a convolutional neural network (CNN) and Lifelong Learning to perform aspect extraction.

In our literature research, we also were informed about some approaches that have used ABAE as their baseline. (Liao et al., 2019) perform unsupervised aspect extraction by using global context and local context to discover aspect word clusters. It proposes using the information of word co-occurrences in the corpus along with

the information of neighboring words at sentence-level. They also use ABAE as a baseline model for their experiments.

(Angelidis and Lapata, 2018) extend the ABAE model by using a few domain-specific keywords for every aspect label as weak supervision.

(Karamanolakis et al., 2019) propose a student-teacher framework that uses seed words for each aspect label, and updates the quality of seed words during model training.

Unlike (Angelidis and Lapata, 2018) who use seed words from a dataset with manual annotations, (Zhao and Chaturvedi, 2019) automatically collect seed words from external information for aspect extraction.

3 Theory

3.1 Autoencoder

An autoencoder neural network is trained to copy its input to its output. Internally, the network may be viewed as consisting of two parts: an encoder function h and a decoder function r defined as:

$$h = f(x) \tag{3.1}$$

$$r = g(h) \tag{3.2}$$

where h is a hidden layer that describes a code that represents the input.(Goodfellow et al., 2016) When the hidden layer h has a smaller dimension than the input dimension, the autoencoder is called undercomplete. The autoencoder is designed to be undercomplete because a hidden layer that learns the input data completely is not especially useful. Instead, autoencoders are restricted with a bottleneck which forces a compressed knowledge representation of the original input.(*Introduction to autoencoders.* 2018) Additionally the network is forced to prioritize which aspects of the input should be copied and learns the most salient features of the input data. The learning process is equivalent to minimizing the loss function given as :

$$L(x, g(f(x))) \tag{3.3}$$

In equation 3.3, L is the loss function that denotes the reconstruction error between the decoder output $g(f(x))$ and the network input x. The two types of loss functions typically used are mean squared distance and Kullback-Liebler (KL) divergence. In the context of this thesis, we limit the description of loss function to the mean squared error function, so the autoencoder objective function can be written as shown in equation 3.4

$$L = arg\,min_{f,g} \|x - g(f(x))\|_2^2 \tag{3.4}$$

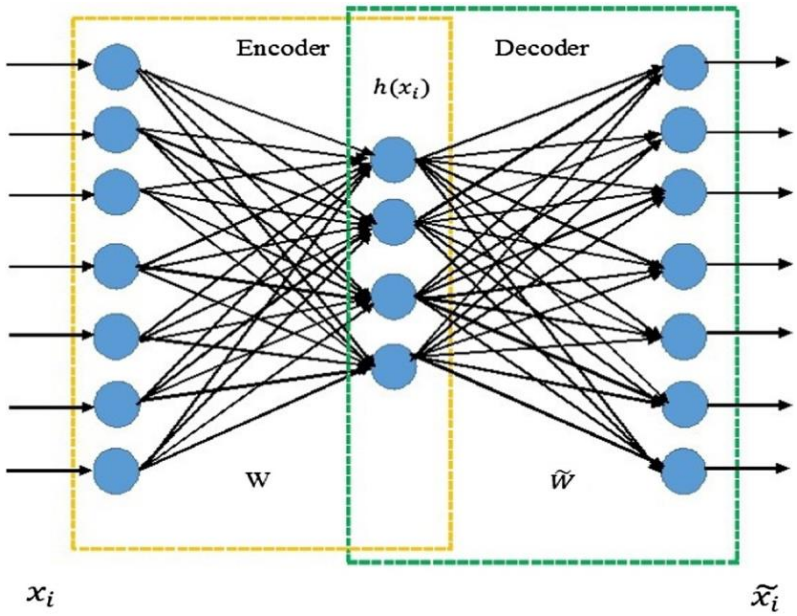


Fig. 3.1.: Source: (Ahmed et al., 2018) Autoencoder architecture

where $\|.\|$ is the L2 norm[1]. The nural network architecture of an autoencoder has been shown in figure 3.1. x_i and $\widetilde{x_i}$ are respectively the input and output layers of the neural network, and $h(x_t)$ is the hidden layer of a smaller dimension. w and $\widetilde{w}$ are weight matrices or parameters of the neural network which are learnt during network training.

3.2 Attention

The principle of attention is to focus only on context-relevant entities and ignore what is not relevant. (Bahdanau et al., 2014) introduced the attention mechanism in the domain of natural language processing. In the research paper, attention is demonstrated using the task of neural machine translation using a recurrent neural network (RNN) based encoder-decoder model which translates an input sequence from source language to target language. The source sentence is vectorized and fed to the encoder RNN; a second decoder RNN in the model architecture translates the input using the last hidden state information of the encoder.(Britz, 2016) To implement the attention mechanism, the decoder output is modified to a weighted combination of all input states from the encoder, instead of just the last hidden state. The weights of each input state are the attention score that defines the importance

[1]L2 norm is a method to compute the length of a vector in Euclidean space, denoted as $\|\| = \sqrt{\sum_i X_i^2}$

of the input state for generating the output. The attention scores are typically normalized using the Softmax function as shown in equation 3.5

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=1}^{T_x} exp(e_{ik})} \tag{3.5}$$

In the equation, e_i measures the alignment of input with the output of the neural machine translation task in (Bahdanau et al., 2014). However, this metric is task specific.

3.3 Topic Modeling

Topic models are probabilistic models that represent the semantic structure in a collection of documents. Topic modeling is a useful tool that uncovers structure in an unstructured collection by connecting documents with similar patterns in data. The latent Dirichlet allocation (LDA) is a majorly used topic modeling method. A *topic* may be defined as a distribution over a fixed number of terms, such that in a collection associated with K topics, each document exhibits all the topics in varying proportions. Topic models are a major unsupervised learning technique in recent research as they perform both aspect extraction and categorization at the same time. Latent Dirichlet Allocation is a traditional topic modeling technique applied to solve the task of aspect extraction. The technique discovers latent topics in text documents, and each topic is treated as an aspect in the context of the task of aspect extraction. Therefore aspect is a distribution over aspect terms. LDA ((Blei et al., 2003)) is an unsupervised generative probabilistic model that groups observations to explain the similarity of data. LDA considers documents as a mixture of latent topics, and each topic is considered a distribution over words.

3.4 Term Frequency–Inverse Document Frequency (tf-idf)

Term Frequency–Inverse Document Frequency (tf-idf) measures the importance of a word within a document in a collection or corpus.(Rajaraman and Ullman, 2011; Vijayarani et al., 2015; Jing et al., 2002) Term frequency may be defined as the number of times a term occurs in a document. Document frequency is the count of documents in which the words appear at least once. The inverse document frequency

is calculated as shown in equation 3.6. The tf-idf is the product of two statistics, term frequency, and inverse document frequency as shown in equation 3.7.

$$IDF(t) = \log \frac{|D|}{DF(t)} \tag{3.6}$$

$$W_i = TF(t_i, d) \times IDF(t) \tag{3.7}$$

In the equation 3.7, W_i is the weight of the word t_i in the document d. The inverse document frequency statistic diminishes the weight of terms that occur very frequently in the document set. This controls the assignment of higher weights to stopwords like 'the', 'a', 'an' which may have high term frequency but don't add significantly to the semantics of the corpus. The tf-idf statistic is often used as a weighting factor in information retrieval and text mining.

3.5 Word2Vec

Word2Vec (Mikolov et al., 2013b; Mikolov et al., 2013a) is a method to create a continuous representation of words or word embeddings. The word embeddings can be trained using two different model architectures: continuous bag-of-words (CBOW) and continuous skip-gram.

Continuous Bag-of-Words (CBOW) : In the continuous bag-of-words architecture, the model predicts the current word from a window of surrounding context words. The architecture is called bag-of-words as the order of words in the context does not influence prediction.

It is further denoted as CBOW, as unlike the BOW model that uses one hot encoding, it uses continuous distributed representation of the context. The dimension of the hidden layer in the neural network is set the same as the target embedding dimension.

Continuous Skip-gram In the continuous skip-gram architecture, the model uses the current word to predict a window of context words. The model is trained as a log-linear classifier, which takes an input word as the current centre word, and predicts the probability of remaining words in the vocabulary to be within the 'context' of the current word.

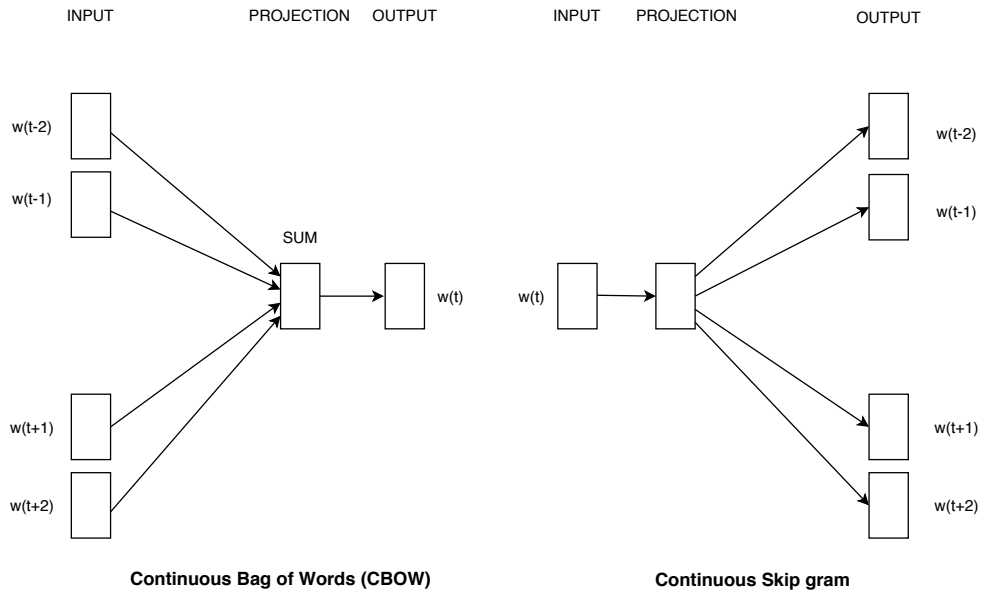


Fig. 3.2.: Graphical representation of the CBOW model and Skip-gram model.

The graphical illustration of the Word2Vec models is shown in figure 3.2.

3.6 GloVe

The Word2Vec models, CBOW and Skip-gram, learn embeddings using local information of a word and a set of words within its context window. The training of Word2Vec models is self-supervised as no labeled data is used for calculating prediction loss. In this learning technique, the statistics of word occurrences in the corpus is the primary source of information available to the model. A claimed drawback of Word2Vec is the under-utilization of corpus statistics, as Word2Vec models train only on separate local context windows, and do not use the information of global co-occurrence counts of words in the corpus. GloVe (Global Vectors)(Pennington et al., 2014) is a model for learning distributed word representations based on the principle that words that co-occur in a context should be semantically connected. The algorithm utilizes global count statistics by constructing a word co-occurrence matrix so that each element of the matrix represents the number of times word w_i appeared in the context of word w_j in the corpus. The training objective of the model is to learn word vectors so that the dot product of word vectors in a word pair is equal to the logarithm of their probability of cooccurrence. As logarithm of a ratio

is equal to the difference of logarithms, the cost function of the model is given as equation 3.8

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \tilde{w_j} + b_i + \tilde{b_j} - \log X_{ij})^2 \tag{3.8}$$

The algorithm uses window size as hyperparameter to define the context window before and after the current word of interest. Distant words are given smaller weights.

3.7 WordNet

The vocabulary of a language may be defined as a set W of pairs (f, s), where a form f is a string over a finite set of alphabets and a sense s is an element from a given set of meanings.(Miller, 1995) A word can have multiple senses or meanings which is interpreted depending on the context of the phrase or sentence where the word is used. When multiple words share a common sense, they are termed as synonyms. WordNet is a lexical database of English. In WordNet, each sense of a word is represented in the form of a synset. An example of a word in WordNet is shown below:

```
word: 'park'
synset: 'park.n.01'
lemma: 'park.n.01.park', 'park.n.01.parkland')
```

Listing 1: An example of WordNet: Synset and lemmas of the word 'park' in WordNet

An example of a word and its synsets are shown in listing 5.3.3

3.8 Predictive Performance Evaluation Metrics

A machine learning model is conventionally trained on a training corpus and evaluated on a held-out dataset or test corpus. A multi-class classification model is evaluated by how accurately it predicts the class or label of data points in an unseen or test dataset. This is measured by assuming the multi-class problem as having only two actual classes, relevant and non-relevant(Manning et al., 2008). The relevant is referred to as positive, and non-relevant classes are referred to as negative. The model's predictions for each class or label can be grouped into four possible outcomes:

		True Class: Positive	True Class: Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Fig. 3.3.: Confusion Matrix of a two-class classification model

True Positive (tp) It is the outcome where the model *correctly* predicts the class of the data point.

True Negative (tn) It is the outcome where the model predicts the *negative or incorrect class* of the data point.

False Positive (fp) It is the outcome where the model *incorrectly* predicts the positive class as the data label.

False Negative (fn) It is the outcome where the model *incorrectly* predicts the negative class as the data label.

The outcomes can be visualized as a two-dimension table which is referred as the **Confusion Matrix**. An example of a confusion matrix is shown in figure 3.3. A confusion matrix provides a detailed overview of the label predictions made by a classification model.

Classification Accuracy

$$accuracy = \frac{tp + tn}{tp + fp + fn + tn} \tag{3.9}$$

Classification accuracy is the fraction of correct predictions made by the classification model.

Precision

$$precision = \frac{tp}{tp + fp} \tag{3.10}$$

Precision is the ratio of correctly predicted positive observations to the total number of predicted positive observations.

Recall

$$recall = \frac{tp}{tp + fn} \tag{3.11}$$

Recall is the proportion of actual positives that were classified correctly.

F1 Score

$$2 \times \frac{precision \times recall}{precision + recall} \tag{3.12}$$

F1 score is the harmonic mean of precision and recall. The highest value of F1 score is 1 and the worst score is 0. For a multiclass classification problem, the harmonic mean of precision and recall can be calculated in two ways:

Macro average F1 score The precision and recall are calculated globally for all the classes.

Micro average F1 score The precision and recall of each label are calculated individually. An unweighted mean of the precision scores of all classes and an unweighted mean of the recall scores of all classes are used to calculate the F1 score. Macroaveraged F1 score gives equal weight to each class and does not consider class imbalance. Whereas micro average F1 score equally weights per-document classification decision,(Manning et al., 2008) and the micro average F1 score is dominated by the major classes in the dataset.

4 Data

Aspect extraction (AE) and aspect-based sentiment analysis (ABSA) tasks are related to the target expressions in a given span of text, and require datasets annotated at aspect level. The annotation procedure is generally done manually and is labor and cost-intensive. Hence there is a blatant shortage in available labeled datasets for this task.

In recent years, researchers have created manually annotated datasets for their experiments and also published the datasets for promoting the progress of the research community. In this thesis, we list six benchmark datasets that have been widely used for the tasks of aspect extraction and aspect-based sentiment analysis. The objective of this overview is to inform about the common conventions of data collection and data annotation in this domain of research. This overview also reflects on some difficulties faced during the dataset construction tasks such as the absence of an explicit aspect term (SemEval-2015), and implicit aspect detection (Sentihood), which aligns with, and reinforces our findings in this thesis.

To construct a list that is relevant to this thesis, we limit our list to datasets only in English and belonging to the domain of social media. The respective descriptions provide insight into:

- *source of data collection, features used for data collection and data statistics*
- *purpose of dataset construction*
- *description of the process of data annotation*

In this chapter, we also discuss the Sentihood dataset, which is used for the experiments in our thesis.

4.1 Datasets

1. **Foursquare dataset**

(Brun and Nikoulina, 2018) published an aspect annotated dataset of restaurant reviews sourced from Foursquare[1], a location platform where users share their experience through comments. The dataset consists of 1006 sentences, collected from 585 user reviews in English about restaurants all over the world. The data points are annotated using the SemEval2016 annotation guidelines for the restaurant domain. The data annotations are performed by a single annotator. The selected annotator is an expert linguist and is also well versed in the SemEval 2016 annotation guidelines. Annotation was performed using BRAT Rapid Annotation Tool[2](Stenetorp et al., 2012). Sentences are annotated with the Opinion Target Expression(OTE), aspect categories, and sentiment polarities. The guidelines followed for aspect category annotation are:

- The entity attribute {E# A} pair defines an aspect (category).
- An entity **E** in an entity aspect {E# A} pair can be assigned to one of the 6 labels *food, drinks, service, ambiance, location, restaurant*. Entities that cannot be described by the set of labels are annotated as *OutOfScope*.
- Attribute **A** of an entity aspect {E# A} pair can be assigned to one of the 5 labels *general, prices, quality, style options, miscellaneous*.
- Annotations are assigned at the sentence level taking into account the context of the whole review.

The dataset is released for use at `https://europe.naverlabs.com/Research/Natural-Language-Processing/Aspect-Based-Sentiment-Analysis-Dataset/`. An example of the annotated data is shown in the listing 2:

```
<text>2 words -filter coffee :-)</text>
-<Opinions>
<Opinion to="22" from="16" polarity="negative" category="DRINKS#QUALITY"
target="coffee"/>
</Opinions>
```

Listing 2: Example of an annotated sentence in the Foursquare dataset

2. **SemEval-2014 Task 4 datasets** The SemEval-2014 Task 4 (Pontiki et al., 2014)dealt with aspect-based sentiment analysis(ASBA). The task involved aspect term extraction, aspect term polarity detection, aspect category de-

[1] `https://foursquare.com/`
[2] `https://brat.nlplab.org/index.html`

tection, and aspect category polarity detection. The tasks are performed on laptops and restaurant review datasets with fine-grained aspect level human annotations. The restaurant data is a subset of the dataset published by (Ganu et al., 2009). The restaurant dataset has 3041 training samples and 800 test samples. The laptops dataset has 3045 training samples and 3845 test samples. The restaurant dataset is annotated with aspect term, aspect term polarity, aspect category, and aspect category polarity. The laptop dataset is annotated with aspect term and aspect term polarity. The aspect categories used for annotation are "*food*", "*service*", "*price*", "*ambience*", "*anecdotes/miscellaneous*". The polarity values used for annotation are "*positive*", "*negative*", "*conflict*", "*neutral*". The annotators used BRAT(Stenetorp et al., 2012) web-based annotation tool. A different annotation process was followed for aspect term annotation and aspect category annotation. For aspect term annotation, each sentence of the two datasets was annotated by two annotators, one of them being an expert linguist. The expert evaluated the annotations done by the second annotator. Annotators followed some guidelines[3] for example: *only explicitly named aspect terms should be tagged, as terms such as* "***everything***" *in* "***everything is noisy***" *do not name any specific aspect.* Disagreements between annotators were reported in multiple categories such as:

- Aspect term detection in sentences with conjunctions or disjunctions consisting multi-word aspect terms such as "*school or office*".

- Distinguishing the aspect term from entity term when opinion target expression is not explicitly mentioned. For aspect category annotation, one annotator validated the existing annotations acquired from the corpus of (Ganu et al., 2009),

Listings 3, 4 are examples of annotated data from the restaurant and laptop dataset for better understanding.

3. **SemEval-2015 Task 12 datasets**

As a continuation of the SemEval-2014 Task 4, two review datasets were released for the SemEval-2015 Aspect Based Sentiment Analysis task (Pontiki et al., 2015). The restaurant dataset has 1315 training sentences and 685 test sentences. The laptops dataset has 1739 training sentences and 761 test sentences. Each sentence is annotated with aspect category and polarity of

[3]The guidelines are available at `http://alt.qcri.org/semeval2014/task4/data/uploads/semeval14_absa_annotationguidelines.pdf`

```
<text>All the appetizers and salads were fabulous, the steak was mouth
watering and the pasta was delicious!!!</text>
<aspectTerms>
        <aspectTerm term="appetizers" polarity="positive" from="8"
        to="18"/>
        <aspectTerm term="salads" polarity="positive" from="23" to="29"/>
        <aspectTerm term="steak" polarity="positive" from="49" to="54"/>
        <aspectTerm term="pasta" polarity="positive" from="82" to="87"/>
</aspectTerms>
<aspectCategories>
        <aspectCategory category="food" polarity="positive"/>
</aspectCategories>
```

Listing 3: Example of an annotated sentence in the SemEval 2014 restaurant dataset

```
<text>From the build quality to the performance, everything about it has been
sub-par from what I would have expected from Apple.</text>
<aspectTerms>
          <aspectTerm term="build quality" polarity="negative" from="9"
          to="22"/>
          <aspectTerm term="performance" polarity="negative" from="30"
          to="41"/>
</aspectTerms>
```

Listing 4: Example of an annotated sentence in the SemEval 2014 laptop dataset

opinions expressed towards the opinion target entities. An aspect category for the datasets includes the entity E and attribute A pair {E# A} towards which the opinion is expressed. Attribute labels used for the restaurant dataset are *'General', 'Prices', 'Quality', 'Style&Options', 'Miscellaneous'*. The attribute labels used for the laptop dataset are 'General', 'Price', 'Quality', 'Operation Performance', 'Usability', 'Design and features', 'Connectivity', 'Portability', 'Miscellaneous'.[4] The polarity labels used are *positive, negative, neutral*. Each dataset was annotated initially by a linguist using the BRAT annotation tool. A second annotator validated/inspected the annotations, and if not sure or not convinced to agree with the annotations, a third annotator was included to make a collaborative decision. (Pontiki et al., 2015) highlight obstacles faced during annotations such as annotation was easier in the restaurant domain, as the number of aspect attributes used for labeling the restaurant dataset was 5, whereas the number of attributes used for labeling the laptop dataset was 8. The annotators also reported that identification of explicit target entity terms and entity terms that were part of a conjunction or disjunction was difficult. Examples of the annotation is shown in the listings 5 and 6 for laptop and restaurant datasets.

[4]The annotation guidelines for SemEval 2015 task 12 datasets are available at `http://alt.qcri.org/semeval2015/task12/index.php?id=data-and-tools`

```
<text>Waited on getting this computer, but it has been a great
change.</text>
-<Opinions>
    <Opinion polarity="positive" category="LAPTOP#GENERAL"/>
</Opinions>
```

Listing 5: Example of an annotated sentence in the SemEval 2015 laptop dataset

```
<text>Great pizza and fantastic service.</text>
-<Opinions>
    <Opinion to="11" from="6" polarity="positive" category="FOOD#QUALITY"
    target="pizza"/>
    <Opinion to="33" from="26"
    polarity="positive"category="SERVICE#GENERAL"
    target="service"/>
</Opinions>
```

Listing 6: Example of an annotated sentence in the SemEval 2015 restaurant dataset

4. **Multi-Aspect Multi-Sentiment(MAMS) dataset**

Jiang et al., 2019 published two versions of the Multi-Aspect Multi-Sentiment(MAMS) dataset for the task of aspect-category sentiment analysis(ACSA) and aspect-term sentiment analysis (ATSA). The ATSA dataset has 11186 training samples, 1332 validation samples, and 1336 test samples. The ACSA dataset has 7090 training samples, 888 validation samples, 901 test samples. The data was collected from the Citysearch New York dataset(cite). For ACSA, eight coarse aspect categories were defined which are *'food', 'service', 'staff', 'price', 'ambience', 'menu', 'place' and 'miscellaneous'*. Three NLP researchers were assigned the task of identifying the aspect categories in the dataset and also determine the sentiment polarities towards these aspect categories. An example of the annotations in the two versions of the MAMS dataset for the task of ACSA and ATSA is shown in the listing 7 and 8. The MAMS dataset was created to challenge the existing ABSA techniques, that have achieved competitive results on existing aspect datasets, with the argument that sentences with a single aspect degenerate ABSA to sentence-level sentiment analysis. Additionally, advanced ABSA models can hardly distinguish varying sentiment polarities towards different aspects expressed in the same sentence. Each sentence in the MAMS dataset consists of at least two aspects with different sentiment polarities. Overall the sentences contain 2.62 aspect terms and 2.25 aspect categories on average.

```
<text>The decor is not special at all but their food and amazing prices make
up for it.</text>
  <aspectCategories>
          <aspectCategory category="ambience" polarity="negative"/>
          <aspectCategory category="food" polarity="positive"/>
          <aspectCategory category="price" polarity="positive"/>
  </aspectCategories>
```

Listing 7: Example of an annotated sentence in the MAMS ACSA dataset

```
  <text>The decor is not special at all but their food and amazing prices
  make up for it.</text>
      <aspectTerms>
          <aspectTerm from="4" polarity="negative" term="decor" to="9"/>
          <aspectTerm from="42" polarity="positive" term="food" to="46"/>
          <aspectTerm from="59" polarity="positive" term="prices" to="65"/>
      </aspectTerms>
```

Listing 8: Example of an annotated sentence in the MAMS ATSA dataset

5. **Twitter Dataset by (Dong et al., 2014)**

(Dong et al., 2014) published a target-dependent Twitter sentiment classification dataset. The training data consists of 6,248 tweets, and testing data has 692 tweets. The data was collected from Twitter using a list of keywords (such as “bill gates”, “taylor swift”, “xbox”, “windows 7”, “google”) to filter comments about celebrities, products, and companies. As shown in the example 4.1, each instance in the dataset is annotated with the target and the polarity label (0: neutral, 1:positive, -1:negative). The target term is replaced with T. Data imbalance was addressed by randomly sampling the tweets. In the resultant dataset, the negative, neutral, positive classes account for 25%, 50%, 25%, respectively. The authors do not mention the precise number of annotators involved in the task, but they report that two annotators were assigned to validate the sentiment annotations done by the authors. The annotators labeled a random sample of tweets and 82.5% of the labels matched correctly with the manual annotations.

```
musicmonday $T$ - lucky do you remember this song ? it ` s awesome . i
     love it .
britney spears
1
they even have red costumes on like $T$
britney spears
0
omg what has the world come to !? they sang $T$ !!!! awful !!!! :
britney spears
-1 }
```

```
cm}
```

Listing 4.1: Annotated sentences in the Twitter dataset by (Dong et al., 2014)

4.2 Sentihood Dataset

The Sentihood dataset published by (Saeidi et al., 2016) was used for the experiments performed in this thesis. The statistics of the dataset are shown in table 4.3. The dataset was constructed for the task of targeted aspect-based sentiment analysis. Each review sentence contains a list of target-aspect pairs $\{t, a\}$ and a sentiment polarity y. The location entity names are masked by location1 and location2. Question-answer pairs specifically related to the neighborhoods within the city of London were collected from the question-answering platform Yahoo! Answers. The location (entity) names that were used for data filtering were taken from the gazetteer GeoNames, and the selection was restricted to reviews that were associated with a location entity name. The content of each question-answer pair was aggregated and split into sentences. Three annotators annotated 10% of the dataset. Each annotator was evaluated by an inter-annotator agreement score measured using Cohen's Kappa coefficient. The annotator with the highest score was selected to annotate all of the datasets. The annotator was not a linguistic expert. A predefined list of aspect labels was provided for annotation given as: live, safety, price, quiet, dining, nightlife, transit-location, touristy, shopping, green-culture, and multicultural. The annotators also reported disagreements in detecting the aspect associated with an expression. As an example provided by (Saeidi et al., 2016), the annotators disagreed on the decision that the phrase "residential area" can be labeled with the aspect "quiet" or "live". An example of an annotated sentence is shown in the listing

```
"opinions": [
    {
        "sentiment": "Positive",
        "aspect": "shopping",
        "target_entity": "LOCATION1"
    }
],
"id": 2013,
"text": " Along LOCATION1 there are lots of Electronics shops (independent
    ones)"
0.5cm}
```

Listing 4.2: Annotated sentences in the Sentihood dataset by (Saeidi et al., 2016)

Dataset	Purpose	Source	number of samples
Foursquare dataset	Aspect based sentiment analysis	Foursquare	1006
SemEval-2014 Task 4 Laptop dataset	Aspect category extraction, Aspect term extraction, aspect term polarity classification, aspect category polarity classification	No information provided	3845
SemEval-2014 Task 4 Restaurant dataset	Aspect category extraction, Aspect term extraction, aspect term polarity classification, aspect category polarity classification	Citysearch New York	3842
SemEval-2015 Task 12 Laptop dataset	Aspect category extraction, Opinion target expression (OTE) extraction, Aspect category polarity classification	No information provided	2500
SemEval-2015 Task 12 Restaurant dataset	Aspect category extraction, Opinion target expression (OTE) extraction, Aspect category polarity classification	Citysearch New York	2000
MAMS ACSA dataset	Aspect category sentiment analysis	CitySearch New York	8879
MAMS ATSA dataset	Aspect term sentiment analysis	CitySearch New York	13854
Twitter Dataset by Dong et al., 2014)	Target based sentiment analysis	Twitter	6940

Tab. 4.1.: List of datasets for aspect category sentiment analysis (ACSA), aspect term sentiment analysis (ATSA), aspect extraction and aspect-based sentiment analysis. For Twitter Dataset by Dong et al., 2014) the task of target based sentiment analysis is equivalent to the task of aspect term sentiment analysis.

Dataset	Method of Annotation	Number of Annotators	Annotation Guidelines	Expert Annotator
Foursquare dataset	BRAT	1	SemEval-2016 Restaurant data annotation guidelines	Yes
SemEval-2014 Laptop dataset	BRAT	2	SemEval-2014 Annotation Guidelines	Yes
SemEval-2014 Restaurant dataset	BRAT	2	SemEval-2014 Annotation Guidelines	Yes
SemEval-2015 Task 12 Laptop dataset	BRAT	1	SemEval-2015 Annotation Guidelines	Yes
SemEval-2015 Task 12 Restaurant dataset	BRAT	1	SemEval-2015 Annotation Guidelines	Yes
MAMS ACSA dataset	Manual	3	None	Yes
MAMS ATSA dataset	Manual	3	None	Yes
Twitter Dataset	BRAT	2	No information provided	No information provided

Tab. 4.2.: Annotation procedure for datasets

Data	Number of Sentences	Labelled	Unlabelled
Train+Dev	3724	2526	1198
Test	1491	1003	488

Tab. 4.3.: Statistics of Sentihood Dataset

5 Methodology and Implementation

The objective of this thesis is to analyze and evaluate unsupervised methods for the task of aspect extraction from the Sentihood dataset. In this chapter, we discuss our motivation towards the three deep learning models used in this thesis. We also discuss the model architectures and the implementation details.

5.1 Methodology

5.1.1 Attention-based Aspect Extraction (ABAE)

Objective The Attention-based Aspect Extraction (ABAE) model is an unsupervised neural model for aspect extraction proposed by (He et al., 2017). The model learns a set of *aspect embeddings* and derives their nearest words in the word embedding space to generate a set of inferred aspects.

Description The ABAE model is analogous to the autoencoder-decoder architecture. An illustration of the model is shown in 5.1. The model architecture can be seen as a combination of an attention mechanism, an encoder, and a decoder. The encoder learns features from the input sentence representations. The decoder reconstructs the sentences via a linear combination of aspect embeddings and the hidden vector.

The words in an input sentence are associated with a feature vector. Word embeddings are used as feature vectors as they map commonly co-occurring words close to each other in the embedding space. The feature vectors form a word embedding matrix $\mathbf{E} \in \mathbb{R}^{V \times d}$, where V is the vocabulary size and d is embedding dimension.

$$d_i = e_{w_i}^T \cdot \mathbf{M} \cdot y_s \tag{5.1}$$

$$a_i = \frac{exp(d_i)}{\sum_{j=1}^{n} exp(d_j)} \tag{5.2}$$

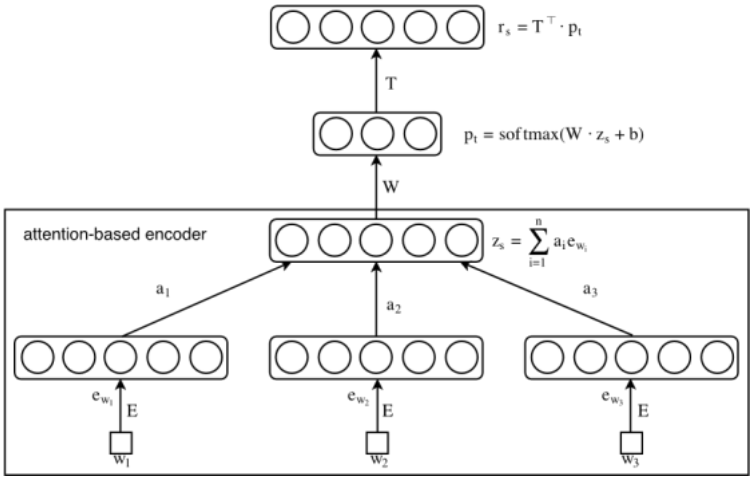


Fig. 5.1.: Source: (He et al., 2017) Architecture of the ABAE model

An attention mechanism is used to down weight and filter non-aspect words in the input sentence. Each word in the input sentence is assigned an attention score a_i computed by an attention mechanism that uses a transformation matrix M, the word embedding w_i and the global context embedding y_s. The matrix M captures the relevance of each word with the global context of the sentence. The global context y_s is derived as the average of all word embeddings. The attention mechanism is represented in the equations (5.2, 5.1).

$$z_s = \sum_{i=1}^{n} a_i e_{w_i} \tag{5.3}$$

The weighted summation of the word embeddings produces a sentence representation z_s used as input to the encoder. (5.3)

$$p_t = \textit{softmax}(W.z_s + b) \tag{5.4}$$

The encoder reduces z_s from d dimensions to an intermediate latent representation p_t of K dimensions, where K is the number of aspect embeddings to be learnt. The latent variable consists information important for subsequent decoding. p_t is a weight vector that represents the probability that the input sentence belongs to

an aspect embedding in the aspect embedding matrix T. p_t is obtained as shown in equation 5.4.

$$r_s = \mathbf{T}^T \cdot p_t \tag{5.5}$$

The decoder then reconstructs the input sentence embedding from a linear combination of the aspect embedding matrix T. (5.5) In the equation, r_s is the reconstructed input representation.

$$J(\theta) = \sum_{s \in D} \sum_{i=1}^{m} max(0, 1 - r_s z_s + r_s n_i) \tag{5.6}$$

The model is trained with a loss function called the contrastive max-margin objective function. The unregularized training objective is given in as 5.6. The function minimizes the reconstruction error between the target sentence embedding z_s and the reconstructed sentence embedding r_s. For each input sample, m sentences are randomly selected from the training data as negative samples n_i. The function is formulated as a hinge loss that maximizes the inner product of r_s and z_s, and concurrently minimizes the inner product between r_s and n_i.

$$U(\theta) = \parallel T_n . T_n^T - I \parallel \tag{5.7}$$

(He et al., 2017) also added a regularization term to ensure that the model learns unique aspect embeddings, and no aspect embeddings learnt during model training is redundant. The regularization term is shown in 5.7, where I is the identity matrix and T_n is the row normalized form of T, such that length of each row is 1. Minimizing the regularization term ensures that the aspect embeddings are orthogonal and hence the dot product between any two different aspect embeddings is zero. The regularized loss function of ABAE is as shown in equation 5.8:

$$L(\theta) = J(\theta) + \lambda U(\theta) \tag{5.8}$$

where λ is a hyperparameter that controls the regularization weight.

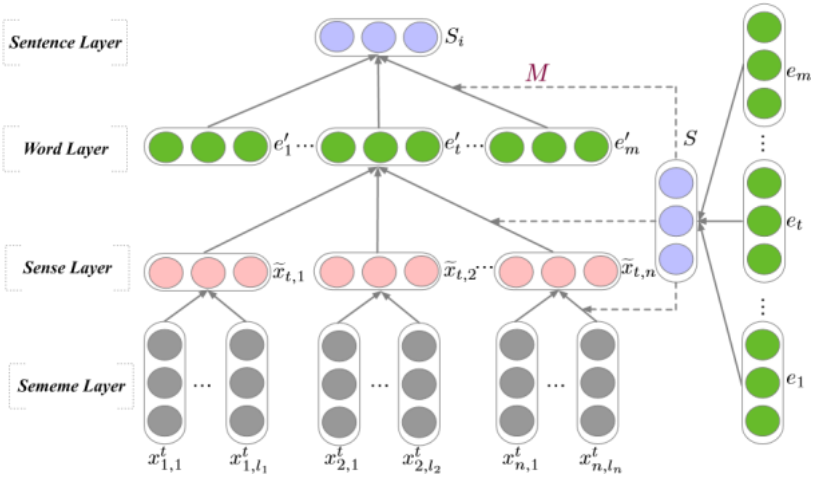


Fig. 5.2.: Source: (Luo et al., 2019) The structure of the hierarchical sememe attention layer

5.1.2 Aspect Extraction with Sememe Attentions (AE-SA)

Aspects in a text can be both explicit and implicit, and implicit aspects are generally present in the form of phrases that do not have a clearly specified aspect term. The extraction of implicit aspects requires the comprehension of the context of the phrase or sentence. To understand a sentence context, it is important to know the correct meaning or sense of the words used in the sentence.

Objective A word can have multiple meanings in multiple contexts. The Aspect Extraction with Sememe Attentions (AE-SA) model by (Luo et al., 2019) addresses the problem by creating distributed representation for a word based on its multiple meanings or senses. The model introduces a hierarchical sememe attention layer (Figure 5.2) that aggregates over multiple word embeddings that might be semantically correlated to the target word. A *sememe* is a minimum semantic unit in a human language, such that each word sense is composed of one or multiple sememes.(Jin et al., 2018) The senses and sememes relevant to a word are obtained from an external lexical knowledge base.

Description The model is analogous to an encoder-decoder architecture. Similar to ABAE, the model learns a set of aspect embeddings that are used to derive their most representative words in the embedding space and create inferred aspects. The overall framework consists of a hierarchical attention mechanism, an encoding, and a decoding module. An illustration of the architecture is shown in figure 5.3.

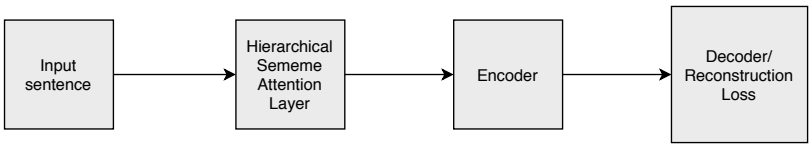


Fig. 5.3.: Illustration of Aspect Extraction with Sememe Attentions (AE-SA) model

Hierarchical sememe attention: The hierarchical attention mechanism assigns weights to the most important senses and sememes for a word depending upon the importance of the word in the context of the sentence. The sentence context is obtained by averaging over the embeddings of the words in the sentence, denoted as E_{avg}.

$$S = \sum_{i=1}^{m} \textit{softmax}(\tanh(e_i^T) \cdot E_{avg}))e_i \tag{5.9}$$

As shown in equation 5.9, each word in the sentence is assigned a weight based on its cosine similarity with E_{avg}. The weight represents the importance of the word in the sentence. An initial sentence representation S is created by taking the summation of the weighted word embeddings.

$$e_t^{'} = \sum_{i=1}^{n} \textit{softmax}(\tanh(\widetilde{x}_{t,i}^T) \cdot S))\widetilde{x}_{t,i} \tag{5.10}$$

A second attention score is calculated to obtain the n senses of the word t included in the initial sentence representation S. The attention score emphasizes the most relevant senses in the context of S. Multiple senses of the t-th word aggregate to form a new word embedding. (5.10).

$$\widetilde{x}_{t,i} = \sum_{j=1}^{l_i} \textit{softmax}(\tanh({x_{i,j}^t}^T) \cdot S))x_{i,j}^t \tag{5.11}$$

The i-th sense of the t-th word in the sentence is the weighted sum of its sememes $\{\mathrm{x}_{i,1}^t, \cdots, \mathrm{x}_{i,l_i}^t\}$. Thus the representation of each word is obtained by using a third attention score that aggregates over its most relevant sememes (5.11).

$$S_i = \sum_{i=1}^{m} \textit{softmax}(\tanh(e_t'^T) \cdot M \cdot S))e_t' \tag{5.12}$$

A fourth attention mechanism filters non-aspect words to obtain a new sentence representation S_i. The attention mechanism uses a trainable transformation matrix $M \in \mathbb{R}$ that captures the most relevant word embeddings in the context of the sentence vector S. (5.12)

$$h = \textit{softmax}(W.S_i + b) \tag{5.13}$$

$$S_o = \mathbf{A}^T.h \tag{5.14}$$

The input representation S_i is compressed by the encoder to a lower dimensional latent variable h (5.13) of dimension K, where K is the number of inferred aspect embeddings to be learnt during model training.

The intermediate representation h is the weight vector over the aspect embedding matrix A, and is used by the decoder to reconstruct the sentence representation by a linear combination of the aspect embeddings. The reconstructed representation S_o is shown in the equation 5.14. As the aspect embeddings and word embeddings share the same embedding space, nearest words to an aspect embedding are used to infer every aspect.

$$J = \sum_{m \in D} \sum_{j=1}^{q} max(0, 1 - S_o^m S_i^m + S_o^m N_j^m) \tag{5.15}$$

$$U = \parallel A_n.A_n^T - I \parallel \tag{5.16}$$

$$L = J + \lambda U \tag{5.17}$$

Similar to ABAE, the model is trained using the contrastive max-margin loss function (5.15) and a regularization term (5.16), where N_j is an average vector representation of the negative samples used for each input sentence. The training objective is shown

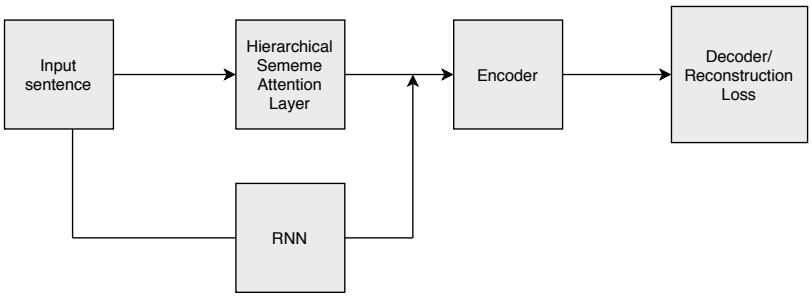


Fig. 5.4.: Illustration of Aspect Extraction via Context-enhanced Sememe Attentions (AE-CSA) model

in the equation 5.17, where λ is a hyperparameter that controls the regularization weight.

5.1.3 Aspect Extraction via Context-enhanced Sememe Attentions (AE-CSA)

The AE-SA model utilizes the multiple meanings an individual word can represent in a sentence. To encode the overall meaning of the sentence, (Luo et al., 2019) introduce the Aspect Extraction via Context-enhanced Sememe Attentions (AE-CSA) model.

Objective: AE-CSA is a variation of the AE-SA model, where it includes an RNN structure that utilizes the sequential relation of words to encode the overall meaning of the input sentence. The AE-CSA model utilizes word sequence information and the lexical semantic information of the words to construct the input sentence representation.
Description:

$$S_i^{'} = \tanh(W^T.(S_i \oplus h_{rnn}) + b) \tag{5.18}$$

The AE-CSA model is an extension of the AE-SA model, with an additional RNN-based structure. An illustration of the architecture is shown in figure 5.4. The original input sentence $s = w_1, w_2 \cdots w_m$ is fed into the RNN to generate hidden representation h_{rnn}. As shown in the equation 5.18, the sentence representation S_i obtained from the hierarchical sememe attention layer and h_{rnn} join to generate a new sentence representation $S_i^{'}$, where $h_{rnn} \in \mathbb{R}^{d^{'}}$, $\oplus$ denotes vector concatenation, $S_i, S_i^{'} \in \mathbb{R}^d$ and d and $d^{'}$ are dimension of the sentence vector and hidden RNN representation respectively. $W \in \mathbb{R}^{(d+d^{'})\times d}$ and b are parameters learnt during training. $S_i^{'}$ is the sentence representation provided as input to the encoder.The

model uses the encoder decoder architecture and the he same objective function as used in the AE-SA model to learn K aspect embeddings.

5.1.4 Evaluation

Cluster Map

"0: 'general', 1: 'transit-location', 2: 'price', 3: 'touristy', 4: 'multicultural', 5: 'shopping'"

The aspect embeddings learned by training a model are used to create inferred aspect clusters. The clusters consist of the top representative words of each aspect embedding, derived using cosine distance metric in the embedding space. The inferred aspects K are mapped manually to a set of gold standard aspect labels predefined for the Sentihood corpus. For model evaluation, a review sentence from the test dataset is assigned to an aspect embedding according to the highest weight in latent vector p_t. Subsequently, the gold standard aspect label mapped to the aspect embedding is the predicted label for the test data. An example of a cluster map is shown in listing 5.1.4

Inferred aspects are manually mapped to gold aspect labels. The inferred aspect with the highest weight assigned by latent vector p_t is assigned as the model prediction. The aspect label mapped to the inferred aspect is the label predicted by the model for the review sentence.

5.2 Baseline Models

A baseline model is the simplest approach to solve a scientific problem, from where we can progressively dive into more complex ones. In the domain of machine learning research, baselines are therefore insightful benchmarks against which we can compare more complex solutions. This enables us to comprehend the ability of our models with higher complexity, and their potential to solve the specified problem. The ABAE, AE-SA, and AE-CSA models are unsupervised aspect extraction methods that primarily detect latent topics in the training dataset. We assess the performance of the models by comparing against two topic modeling baselines: K-means and Non-Negative Matrix Factorization (NMF).

5.2.1 K-means

The ABAE model is a neural network architecture whose objective is to learn a set of K aspect embeddings, such that the word embeddings used to represent the model input and aspect embeddings learned as a model parameter share the same embedding space. The aspect embeddings are learned in the form of an aspect embedding matrix $T \in \mathbb{R}^{K \times d}$, where d is the dimension of word embeddings. K-means algorithm is applied on word embeddings to obtain K cluster centroids which are then used to initialize the aspect embedding matrix T. K-means is an unsupervised clustering method that separates a set of N samples into k disjoint groups or clusters where $k \leq N$. The samples are assigned to the clusters so that the inter-point distances are small compared with the distances to points outside the cluster. (Bishop, 2006) Each cluster is represented by a centroid μ_k, so the objective function[1] or *distortion measure* can be formally written as shown in equation 5.19.

$$\sum_{i=0}^{n} \min_{\mu_j \in C} (\|x_i - \mu_j\|^2) \tag{5.19}$$

The equation 5.19 represents the within-cluster sum-of-squares criterion, where n is the set of samples to be clustered, and C is the number of clusters to be learned with the data. Similar to the expectation-maximation algorithm (Moon, 1996) the algorithm iterates through two steps: 1) Each sample is assigned to its nearest centroid. 2) Mean value of all samples in a cluster is assigned as the new cluster centroid. The algorithm iterates amongst the two steps until the distance between old centroids and new centroids is below a threshold value, that is there is no further change of centroid assignments. We use K-means as a baseline model to compare the coherence of clusters generated by the ABAE model and to gauge the potential of ABAE. Similar to ABAE, K-means is applied on the word embedding matrix to obtain cluster centroids which are d dimensional vectors in the word embedding space. Cosine similarity of the centroids and the word embeddings are used to retrieve the top x representative words for each centroid in the word embedding space. The top representative words of the centroids form the inferred aspects of the baseline model.

[1] K-means in ABAE model and baseline model are applied using the implementation provided by scikit- learn which is a machine learning library for Python programming language. `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.K-means.html`

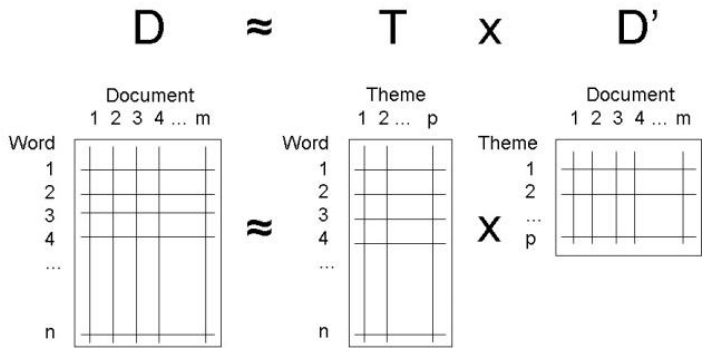


Fig. 5.5.: Source: (Bravo-Alcobendas and Sorzano, 2009) Schematic representation of the matrix decomposition performed by NMF.

5.2.2 Non-Negative Matrix Factorization (NMF)

The Non-Negative Matrix Factorization (NMF) algorithm is an unsupervised technique to reduce the dimensions of non-negative matrices.(Belford et al., 2018) Given a non-negative matrix $X \in n \times m$, NMF factorizes it into two nonnegative matrices W and H, such that $X \approx WH$, where W is a $n \times k$ matrix and H is $k \times m$ matrix. Typically k is much smaller than m and n. A schematic illustration of NMF is shown in figure 5.5.(Bravo-Alcobendas and Sorzano, 2009)

When NMF is used as a topic extraction baseline for ABAE, X is the term-document matrix formed by the Sentihood training corpus. Each review sentence in the corpus is a 'document', and the constituent words in a document are denoted as 'terms'. NMF is used to factorize X into two lower dimension matrices W and H, so that the rows of the factor H can be interpreted as k topics and the columns of H define their non-negative weights for each of the m terms in the corpus vocabulary. Similarly, the columns of factor W provide membership weights for all documents with respect to each of the k topics. The NMF is then an optimization problem whose objective function(Pedregosa et al., 2011) is the squared Frobenius norm as shown in equation 5.20:

$$d_{Fro}(X, Y) = \frac{1}{2}\|X - WH\|^2_{Fro} \tag{5.20}$$

where $\|.\|$ is the Frobenius norm. The objective function is the distance between X and the matrix product of W and H, subject to the constraints W, H ≥ 0.

5.3 Implementation

In this section we look into the implementation details for each model. We also list the constraints and limitations faced during implementation and the workaround applied to perform the experiments.

5.3.1 Data Preprocessing

The Sentihood dataset was subjected to a preprocessing pipeline to make it analyzable and predictable for the task. We used the implementation by (Ma et al., 2018) to preprocess the Sentihood dataset. As the review sentences are short in length, modifications were kept to be minimum to preserve context.

Lowercasing: All sentences are lowercased to prevent mixed-case occurrences of words so that words like 'england' and 'England' have the same representation in the word embedding space.

Tokenization : The sentences were tokenized using the Scikit-Learn's tokenizer. Tokenization is the process of splitting words into individual tokens. The Scikit-Learn's tokenizer also filters all singularly occurring numbers and letters from the dataset.

Stop word removal: Stop words were removed from the corpus using NLTK (Natural Language Toolkit)[2] which is a library widely used for statistical natural language processing.(Loper and Bird, 2002). Stop words are high-frequency words in a language such as 'the', 'is', 'are', and are presumed to add less lexical value to a given text. Generally, they are filtered before an NLP task to save space and processing time. There is no universal list of stop words and we use the list of English stopwords provided by NLTK. A detailed list of stopwords is provided in the appendix.

Lemmatization: Each sentence was lemmatized to remove inflectional endings of words that may be visible in the text such as 'run' and 'runs'. Lemma is the base or dictionary form of a word. The NLTK API for the WordNet lemmatizer was used for lemmatization. It returns the input word unchanged if it cannot be found in WordNet. An example of the preprocessing pipeline is shown in the listing 5.3.1.

[2] `https://www.nltk.org/`

5.3.1 Preprocessing of Sentihood dataset

Text:
"LOCATION1 has been gentrified for over 40+ years, using London incomes to push property prices even higher than the national average, because of it's proximity to central London'

Step 1. Lowercasing & Tokenization
['location1', 'has', 'been', 'gentrified', 'for', 'over', '40', 'years', 'using', 'london', 'incomes', 'to', 'push', 'property', 'prices', 'even', 'higher', 'than', 'the', 'national', 'average', 'because', 'of', 'it', 'proximity', 'to', 'central', 'london']

Step 2. Stopword removal
['location1', 'gentrified', '40', 'years', 'using', 'london', 'incomes', 'push', 'property', 'prices', 'even', 'higher', 'national', 'average', 'proximity', 'central', 'london']

Step 3: Lemmatization
['location1', 'gentrified', '40', 'year', 'using', 'london', 'income', 'push', 'property', 'price', 'even', 'higher', 'national', 'average', 'proximity', 'central', 'london']
Here 'using' and 'higher' are returned unchanged as we did not use parts of speech for lemmatization.

Preprocessed data point:
location1 gentrified 40 year using london income push property price even higher national average proximity central london

5.3.2 ABAE

We have used the implementation of ABAE model published by (He et al., 2017) for our experiments. A vocabulary dictionary was created with the unique words in the training corpus. The word embeddings of the words in the vocabulary were stored in the embedding matrix E. K-means was applied on E to obtain a set of cluster centroids. These centroids were used to initialize the aspect embedding matrix T, which is a parameter learned during model training. For all experiments, the ABAE model was trained on the training and development dataset of the Sentihood corpus, without the labels as we only evaluated unsupervised approaches. Word embeddings are kept fixed during training.

5.3.3 AE-SA

Example of senses and sememes in WordNet

Word in a sentence: *'wow'*

Senses/Synsets available in WordNet:
[Synset('belly_laugh.n.02'), Synset('wow.v.01')]

Sememe of synset 'belly_laugh.n.02':
[Lemma('belly_laugh.n.02.belly_laugh'), Lemma('belly_laugh.n.02.sidesplitter'), Lemma('belly_laugh.n.02.howler'), Lemma('belly_laugh.n.02.thigh-slapper')]

All sememes for all senses related to each word in a review sentence were obtained from Wordnet which is a large lexical database for English. WordNet originally defines a synset in terms of *lemmas*. The term *sememe* is used to describe the smallest lexical units in the Chinese lexical database HowNet(Dong et al., 2010). We mapped the concept of *sememes* in HowNet to the concept *'lemma'* in WordNet, as in both the databases they compare equivalently as the smallest semantic unit. Thus in this thesis, the term *sememe*' is equivalent to the term *'lemma'*. An example of the synset and sememes of a word in WordNet is shown in the listing 5.3.3. The list of sememes is cropped in the example for the sake of keeping the description brief.

To construct the input representation in the AE-SA model, word embeddings of each word sense and the corresponding sememes were extracted from the word embedding matrix E to create an input sentence representation which is then fed to the encoder. The Sentihood dataset consists of misspelled words, nouns, and alphanumerical words that are not part of the English vocabulary and hence are unavailable on WordNet. For these exceptions, we used the original word embedding itself.

5.3.4 AE-CSA

The existing implementation of the AE-SA model was used as the model architecture. An LSTM module with a hidden size 500 was used as the required RNN structure and was added to the existing AE-SA model.

5.3.5 K-means

The K-means algorithm was used to initialize the aspect embedding matrix in all the neural models. Therefore as a baseline, it is applied to obtain k centroids from the word embeddings matrix. These centroids form the aspect embedding matrix for the baseline model, and no further model training is performed. For each aspect embedding, cosine similarity is used to obtain the top 100 representative words in the word embedding space, which then form the inferred aspect clusters.

5.3.6 NMF

The sentences in the Sentihood training corpus were transformed into Term Frequency-Inverse Document Frequency (TF-IDF)vectors and converted to a document term matrix. The top representative words obtained from the topic word matrix form the inferred aspect clusters. We used the NMF implementation provided by Scikit-Learn[3](Pedregosa et al., 2011) which factorized the matrix into two lower rank matrices. The topic term matrix was used to obtain the top 100 words for each topic and create a list of inferred aspects.

[3]The NMF implementation provided by Scikit-Learn is available at `https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.NMF.html`

6 Experiments

In this chapter, we outline the experiments performed with the ABAE, AE-SA, and AE-CSA models. The procedure of an experiment with the ABAE, AE-SA, and AE-CSA models includes **a)** model training and **b)** model evaluation with a human in the loop. We distinguish our experiments as transfer learning and deep learning.

> ***Transfer learning*** As ABAE, AE-SA and AE-CSA are unsupervised deep learning models, experiments that utilize pre-trained word embeddings are identified as transfer learning. In this thesis, we perform 7 experiments based on transfer learning.
>
> ***Deep learning*** Experiments with word embeddings trained on the Sentihood corpora are categorized as deep learning. In this thesis, we perform 2 experiments based on deep learning.

We also conduct experiments with K-means and NMF which are our baseline models.

Word Embeddings Word2Vec and GloVe are two-word embedding methods that are widely used in NLP tasks. (He et al., 2017) used Word2Vec trained on restaurant and laptop review datasets for input representation in the ABAE model. As we perform our experiments on a different dataset, we explore a second word embedding method GloVe, as different embedded vectors may highlight different syntactic features of the corpus. In the following sections, we describe the details for training word embeddings, along with the experiments performed in this thesis.

Grid Search Hyperparameters are used to control the process of model training. In our experiments, aspect size is a hyperparameter that is optimized to obtain the most coherent clusters and maximum classification accuracy in the task of aspect label prediction. The aspect size is equivalent to the number of inferred aspects the ABAE model should detect in the training corpus in an experiment. We initially applied heuristic search or human-guided search and followed with a conventional grid search. Experiments were performed as grid search for aspect count 10 to 100, with an interval of 10.

Hyperparameters	Values
Vocabulary size	9000
Learning rate	0.001
Epochs	15
Batch size	32
Negative samples per input sample	20
Regularization weight	0.1
Embedding dimension	100

Tab. 6.1.: Hyperparameters for aspect extraction task

Model training hyperparameters We applied the hyperparameters used by (He et al., 2017) for all our experiments. A list of the hyperparameters is given in table 6.1.

6.1 Transfer Learning

Pre-trained word embeddings Word embedding models need a large corpus of text for training and generating informative representation vectors. As the size of the Sentihood dataset is comparatively small for deep learning experiments, we initialize the word embedding matrix with pre-trained GloVe embeddings that were trained on other large text corpora. We used pre-trained GloVe embeddings available at the Stanford NLP Github repository[1] (Pennington et al., 2014) which were trained on Wikipedia[2] 2014 which had 1.6 billion tokens, and Gigaword 5 [3] (Parker and Linguistic Data Consortium, 2011) corpus which had 4.3 billion tokens. Wikipedia is a free online encyclopedia and Gigaword is a comprehensive archive of newswire text data in English acquired over several years by the Linguistic Data Consortium (LDC). The embeddings were trained on a vocabulary size of 400000 words. The repository provides other pre-trained GloVe embeddings that are trained on Common Crawl and Twitter, but we used the embeddings trained on Wikipedia and Gigaword, as we expected Wikipedia to have geographical and historic data regarding London, and Gigaword 5 , which is based on multiple newswire sources, to have text information regarding the city or an overview of its urban neighborhood which is the domain of Sentihood dataset.

[1] `https://nlp.stanford.edu/projects/GloVe/`

[2] `https://en.wikipedia.org/wiki/Main_Page`

[3] The text in Gigaword 5 comes from seven distinct international sources of English newswire: Agence France-Presse, Associated Press Worldstream, Central News Agency of Taiwan, Los Angeles Times/Washington Post Newswire Service, Washington Post/Bloomberg Newswire Service, New York Times Newswire Service, Xinhua News Agency.

6.1.1 ABAE

1. **Experiment with pre-trained GloVe embeddings** We initially applied a heuristic approach based on the proportion of the number of gold standard aspect labels and the number of inferred aspects. As shown by (He et al., 2017), the maximum cluster coherence was obtained at a count of 14 inferred aspects, and the number of gold standard aspects for the restaurant dataset was 6. Hence we used the 2:1 ratio and performed our experiments with 14, 28, 37, and 42 inferred aspects. The hyperparameters were kept uniform with other experiments are shown in the table 6.1

2. **Experiment after removing low precision classes** We observed that the highest classification accuracy is received in the ABAE experiment with pre-trained GloVe embeddings for 37 aspects (shown in section 7). However, the classification metrics of this experiment and all the preceding experiments for aspects 10, 14, 28, and 37 displayed a pattern of zero precision and recall scores for several aspect classes. (Here aspect classes refer to the predefined gold standard aspects of the Sentihood dataset (Saeidi et al., 2016).) This was related to imbalanced class distribution (section 7) in the experimental dataset, low frequency, and implicit aspects which are difficult to identify even by a human observer. This is discussed in more details in section 7.1.1.1. A selective number of aspects which are majorly represented in the dataset were observed to obtain higher precision and recall scores. To observe the accuracy of the model in predicting only this selective list of aspect classes, we conducted a new experiment where we filtered the classes with low or null precision and recall scores. The classes were removed from the experiment by

    - removing the test set data points, whose true label belonged to the set of omitted gold standard aspect classes.

    - no inferred aspect was mapped to any of the omitted classes during cluster mapping. If an inferred aspect representative words were highly related to one of the specified classes, the former was mapped to the general class.

    Hence, none of the labels from this list was assigned as a predicted label by the trained ABAE model during model evaluation. The list of classes removed from the experiment was *'dining','green-nature','green-culture','live','quiet'* and *'touristy'*.

3. **Experiment with embeddings finetuned to Sentihood corpus** The Sentihood dataset has only 3793 words in the vocabulary, and hence is a very small corpus for learning input features using deep learning architectures. We hoped that finetuning the pre-trained GloVe representations to our dataset might improve the coherence of the inferred aspects. Mittens (Dingwall and Potts, 2018) is a python package that finetunes pre-trained GloVe representations to a specialized domain. The training hyperparameters used are the embedding dimension and number of iterations. For our experiment, we used the default value for the number of iterations that is 1000, and the embedding dimension was set as 100. The embedding matrix of the ABAE model was initialized with the finetuned pre-trained GloVe embeddings for model training. The experiment was performed only for aspect size 37, which obtained the highest classification performance and was used as a baseline score for our experiments.

4. **Experiment with embeddings finetuned to Sentihood corpus and without low precision classes** In a previous experiment, we initialized the embedding matrix of the ABAE model with pre-trained GloVe embeddings finetuned to the Sentihood corpus. We repeat the experiment with the same hyperparameters inclusive of aspect count, but we omit some classes which had a null precision score in the previous experiments. The list of classes removed from the experiment was *'dining', 'green-nature', 'green-culture', 'live', 'quiet'* and *'touristy'*. The process of removing low precision classes is the same as discussed in experiment 2. This experiment was performed for the sake of completion, as well as to measure the accuracy of the trained model in predicting a selective set of aspect labels with high precision and recall scores.

5. **Experiment after removing gibberish data** During the qualitative evaluation (7) of inferred aspects obtained in the ABAE experiment with pre-trained GloVe embeddings, we observed experiments with a higher number of inferred aspects generated a higher number of coherent clusters, compared to experiments with a smaller number of inferred aspects where the clusters were majorly incoherent. (For a detailed discussion refer to experiment 1) During the analysis, we observed that in experiments with a higher number of inferred aspects, several clusters consist of nouns, three letter words, unstructured words, and alphanumerical. These words don't belong to the English language and are also difficult to be manually mapped to one of Sentihood's predefine gold standard aspects. To increase the number of coherent aspects, we filtered non-English and alphanumerical words from the Sentihood training corpus and trained the ABAE model on the cleaned Sentihood training corpus. The inferred aspect count for the experiment was 37, as in previous experiments

we obtained the best classification accuracy at this aspect size. The training hyperparameters were kept the same as the rest of the experiments.

6.1.2 AE-SA

In this experiment, we initialize the word embedding matrix of the AE-SA model with GloVe embeddings pre-trained on Wikipedia 2014 and Gigaword 5 corpus. The model is then trained on Sentihood's training dataset. The word embeddings were not retrained during model training. Grid search was performed by experimenting with aspect count 10 to 100 with an interval of 10. The objective of grid search is to find the optimal aspect size where the model generates inferred aspects with maximum cluster coherence.

6.1.3 AE-CSA

The AE-CSA model was trained on Sentihood's training dataset. The word embedding matrix was initialized by GloVe embeddings pre-trained on Wikipedia 2014 and Gigaword 5 corpus. The word embeddings were not retrained during model training.

6.2 Deep Learning

1. **ABAE with Word2Vec** In this experiment, we initialize the word embedding matrix with Word2Vec embeddings trained on the experimental dataset. The embeddings were trained using the Word2Vec Continuous Bag-of-Words (CBOW) algorithm as was used by (He et al., 2017). We used the Word2Vec CBOW implementation [4] provided by Gensim. Gensim is a library focused on topic modeling and popular implementation of the Word2Vec algorithm. For training the embeddings, we utilized the hyperparameters used by (He et al., 2017) and set the window size to 10 and a negative sample size to 5. So we don't generate embeddings for all words with a total frequency lower than the hyperparameter min_count. We set the embedding size to 100, as we use this embedding size for training GloVe embeddings as well, and we want to use a uniform dimension in all embedding models. The saved embeddings are loaded in the ABAE model as *keyedvectors* to reduce RAM footprint. *'KeyedVectors'* is a structure defined by Gensim as a mapping between entities and vectors, where entities may correspond to words, documents, etc. The embeddings

[4]The Word2Vec CBOW implementation provided by Gensim is available at https://radimrehurek.com/gensim/models/Word2Vec.html

were trained in two phases. In the first phase, embeddings were trained only on the training data split. To improve the low classification accuracy observed in the consecutive experiments, the second set of embeddings were trained jointly on the training and development data set. From hereafter, we refer to the joined training and development dataset as the training dataset.

Hyperparameters	Values
VOCAB_MIN_COUNT	5
WINDOW_SIZE	15
MAX_ITER	50
VECTOR_SIZE	100
NUM_THREADS	8
X_MAX	10

Tab. 6.2.: Hyperparameters for training GloVe embeddings on Sentihood dataset

2. **ABAE with GloVe embeddings trained on Sentihood corpus** The GloVe embedding model was trained on the Sentihood dataset for obtaining vector representations of the words in the corpus. We use the algorithm implementation provided by (Pennington et al., 2014)[5] for training the embeddings. We used the hyperparameters predefined in the implementation which are VOCAB_MIN_COUNT, WINDOW_SIZE, and MAX_ITER. VOCAB_MIN_COUNT was set as 5, so no embeddings were generated for words occurring less than 5 times in the corpus. The vocabulary size for the Sentihood dataset is 3796, and keeping a VOCAB_MIN_COUNT equivalent to 5 generated word embeddings for 829 words only. Although it is a huge reduction in the coverage of the dataset, we keep the hyperparameters as default so that we can compare the results from our aspect extraction experiments across models using different embedding vectors. The hyperparameters are shown in table 6.2. Two different sets of embeddings were trained in two separate phases. The first training was conducted with the hyperparameter MAX_ITER as 15 which is the default value in the implementation available at the Stanford NLP Github repository. In a second additional phase, a new set of word embeddings were trained on the same corpus with the hyperparameter MAX_ITER set as 50. This was done to observe any improvements in classification accuracy in the following experiments. The embedding dimensions were chosen as 100 for all experiments.

6.3 Baseline

[5]The GloVe code is available at `https://nlp.stanford.edu/projects/GloVe/`

6.3.1 K-means

This experiment was conducted as a baseline for the transfer learning experiments discussed in a previous section (6.1). The embedding matrix was initialized with GloVe embeddings pre-trained on Wikipedia 2014 and Gigaword 5 corpus. As the model is not trained on the training corpus, we did not use any model training hyperparameters in this experiment.

6.3.2 NMF

The NMF implementation provided by Scikit-Learn was used to factorize the training corpus. The 'n_components' parameter is equivalent to aspect size and was altered for obtaining topic term matrix for a range of 10 to 100 topics, at an interval of 10.

6.4 Evaluation

Inferred Aspects	Representative Words	Gold Standard Aspect
Real estate worth	dollar, worth, equates, upwards, priced, rocketed, estimate	Price
City administration	prime, party, lebanese, movement, thai	Multicultural
Movement zones	parked, embankment, flooded, nearby, tunnel, parking, surrounded	Transit-Location
Royal family	lord, kingdom, king, abolished, viii, elizabeth, empire	Touristy
Brands	company, ikea, tesco, giant, kfc, ebay, bought, sale, buy, chain	Shopping
Entertainment	circus, cabaret, entertainer, comedy, concert, nightclub, musical	Nightlife

Tab. 6.3.: List of inferred aspects (left), subset of representative words used for cluster topic decision(middle), and the mapped gold standard label(right).
The predefined aspect labels for Sentihood dataset are *'live', 'safety', 'price', 'quiet', 'dining', 'nightlife', 'transit-location', 'touristy', 'shopping', 'green-culture', 'multicultural', 'general'*

The inferred aspects obtained in an experiment were manually mapped to the gold standard labels predefined for Sentihood corpus.(Saeidi et al., 2016). The Sentihood corpus has 12 gold standard labels: *'live', 'safety', 'price', 'quiet', 'dining', 'nightlife', 'transit-location', 'touristy', 'shopping', 'green-culture', 'multicultural', 'general'*. A list of representative words and the mapped gold labels are shown in the table 6.3. The cluster mapping was done on a one-to-one basis and to have as many labeled clusters

as possible to improve the classification accuracy for the associated labels. Hence we avoided putting clusters in the general category. As an example, adverbs like *'cheerfully'* were mapped to initially to the label 'live', and then to general for later experiments. Also in contradiction to the general assumption that the top 50 words represent the cluster more strongly if a few significant words appeared after the top 50 words, they were considered important for mapping. As no instructions were available for cluster interpretation, apart from general human knowledge, a cue was taken from the original aspect annotations available in the Sentihood dataset. This was not thoroughly used for all models and all experiments, but only for the initial understanding of the human observer. The clusters that consisted of a combination of multiple topics were randomly mapped to one of the suitable gold labels. Similarly, ambiguous clusters were homogeneous but consisted of words that could be mapped to multiple gold labels. As an example, a cluster that includes words like *'10min', 'million', 'central'* can be inferred as *'price'* as well as *'transit-location'*.

7 Results & Discussions

In this chapter, we assess the results of our experiments with the ABAE, AE-SA, and AE-CSA model. We also report the results obtained for our baseline models. The results consist of the inferred aspects generated by the models, and the classification metrics obtained by evaluating the trained models on the Sentihood test dataset. An experiment may be defined as training a model on the Sentihood dataset for a specific aspect size, followed by evaluation. Therefore, we analyze each experiment in terms of the coherence quality of the inferred aspects (quantitative analysis), and the classification performance on the evaluation dataset.

We further discuss the observations and interpretations of the human in the loop, who is responsible for the cluster mapping in each experiment. This section is termed as qualitative analysis.

We follow a similar organizational structure as used in the experiments chapter for visual alignment. Hence our results are listed according to the two types of approach we have utilized in our experiments (transfer learning, deep learning), followed by the results obtained from our baseline models.

Quantitative analysis of inferred aspects:
A coherent topic is one that uniformly speaks about one subject. For example {*'November','December','August'*} is a cluster of three words that tell about the topic *'months'*. In contrast, a cluster of words {*'apple', 'sand', 'money'*} does not convey one uniform topic. Topic coherence is an intrinsic evaluation of the semantic nature of topics learned in topic modeling.

We measure the quality of the inferred aspects obtained in our experiments using the UMass coherence score (Mimno et al., 2011) and the Word Embedding Based Topic Coherence (Ding et al., 2018). The metrics are used to evaluate whether the generated topics are interpretable by humans. A higher score indicates a more meaningful and semantically coherent topic. As observed in our experiment results, coherent topics are the ones where all representative words associate with a common theme.

UMass coherence metric: The UMass coherence metric is based on word co-occurrence statistics within the documents being modeled. It is based on the assumption that words belonging to a single concept are more likely to cooccur in a document, compared to a pair of words that belong to different or unrelated concepts. Given an aspect z and set of its top N words, $S^z = \{w_1^z, \cdots, w_N^z\}$, the coherence score is calculated as in equation 7.1

$$C(z; S^z) = \sum_{n=2}^{N} \sum_{l=1}^{n-1} \log \frac{D_2(w_n^z, w_l^z) + 1}{D_1(w_l^z)} \tag{7.1}$$

where $D_1(w)$ is the document frequency of word w, and $D_2(w1, w2)$ is the co-document frequency of the words w_1 and w_2. The metric corresponds well with human coherence judgments.

Word Embedding Based Topic Coherence (WETC): We use a second coherence metric called word embedding based topic coherence (WETC). The metric calculates the coherence of a topic cluster using the pair-wise cosine similarity of the cluster members, in the word embedding space. As shown in equation 7.2, E is the word embedding matrix for a list of N words, such that $E \in^{N \times D}$, D is the embedding dimension, and $\langle ., . \rangle$ denotes matrix product.

$$WETC_{pw}(E) = \frac{1}{N(N-1)} \sum_{j=2}^{N} \sum_{i=1}^{j-1} \langle E_{i,:}, E_{j,:} \rangle \tag{7.2}$$

To calculate the coherence score for an experiment with a model and an aspect size, we calculate the coherence for each inferred aspect obtained in the experiment. The mean of all coherence scores is taken as the coherence score of the experiment.

Qualitative analysis of inferred aspects:
As our experiments involved human supervision to map inferred aspects to gold standard aspects pf the Sentihood dataset, we use the observer's understanding of the aspects as a second technique of analyzing the model performance. This is considered because manual cluster mapping was difficult for models with incoherent clusters and easier for models with improved coherence.

Classification metrics:
We have used Scikit-Learn's classification report to visualize the performance of each model for the task of predicting the gold standard aspect label for the evaluation dataset. Following the work of (Angelidis and Lapata, 2018) we use micro F1 score and accuracy to evaluate classification results of our models.

Aspect	ABAE	K-means	AE-SA	AE-CSA	NMF	ABAE_SH	Sum of AUC
10	-148.874.201	-151.909.810	-145.356.715	-142.837.322	-104.959.758	-3.563.630	**-697.501.436**
20	-149.321.800	-154.448.616	-144.587.619	-146.663.804	-115.294.879	-5.523.148	-715.839.866
30	-150.333.167	-154.371.669	-146.056.441	-148.641.333	-120.255.853	-4.151.290	-723.809.753
40	-150.279.860	-155.410.926	-147.670.604	-149.235.099	-124.179.916	-3.099.355	-729.875.760
50	-150.610.692	-155.123.895	-148.138.198	-149.853.505	-127.195.119	-2.794.886	-733.716.295
60	-150.894.758	-155.509.730	-148.570.454	-149.694.998	-128.239.363	-2.394.719	-735.304.022
70	-151.183.497	-155.250.056	-149.043.904	-150.081.084	-129.863.126	-1.999.210	-737.420.877
80	-151.034.824	-155.081.512	-149.003.840	-150.061.577	-131.019.218	-1.728.078	-737.929.049
90	-151.236.573	-155.231.713	-149.284.444	-150.679.502	-132.296.331	-1.569.474	-740.298.037
100	-151.183.658	-155.316.483	-149.268.989	-150.320.469	-132.354.901	-1.409.682	-739.854.182
Sum of AUC	-1.504.953.030	-1.547.654.410	-1.476.981.208	-1.488.068.693	-1.245.658.464	**-28.233.472**	

Tab. 7.1.: Quantitative analysis: Area under curve(AUC) for all models for inferred aspect cluster coherence calculated using the UMass Coherence Metric, for aspect count 10 to 100. Lower value is better. Aspect 10 achieves the highest AUC in the list of aspect counts. ABAE-SH achieves the highest AUC in the list of all models

7.1 Results

The tables 7.1 and 7.2 show the coherence scores achieved by each model for aspect count 10 to 100. The models compared are ABAE, ABAE_SH, AE-SA, AE-CSA, NMF and K-means. Table 7.3 lists the classification accuracy and microaveraged F1 score obtained for all experiments.

It is to be noted that as the AUC scores obtained for the Umass coherence values are all negative, a lower AUC value denotes a higher coherence score and a more coherent topic model (a higher value with the negation symbol signifies a smaller coherence). The ABAE_SH model obtains the most coherent clusters across all experiments. In the AUC score table for the WETC metric, a higher score denotes higher cluster coherence (as all scores are positive real numbers). The AE-SA model obtains the most coherent clusters across all experiments.

Aspect	ABAE	K-means	AE-SA	AE-CSA	NMF	ABAE_SH	Sum of AUC
10	9.535	9.264	11.133	11.816	15.700	16.022	73.470
20	13.414	8.753	15.550	13.230	11.642	14.673	**77.262**
30	13.467	9.488	14.483	12.832	12.242	11.639	74.151
40	13.553	9.252	15.332	13.073	10.342	9.770	71.322
50	12.394	9.583	14.565	11.503	10.087	7.968	66.100
60	12.493	9.596	14.580	11.797	9.203	7.589	65.258
70	12.822	9.794	14.698	12.566	9.132	7.742	66.754
80	12.573	9.456	14.434	12.590	8.651	6.978	64.682
90	12.483	10.325	14.599	11.089	8.820	6.598	63.914
100	11.490	9.752	14.546	10.821	9.136	6.619	62.364
Sum of AUC	124.224	95.263	**143.920**	121.317	104.955	95.598	

Tab. 7.2.: Quantitative analysis: Area under curve(AUC) for all models for inferred aspect cluster coherence calculated using the WETC Metric, for aspect count 10 to 100. Higher value is better. Aspect 20 achieves the highest AUC in the list of aspect counts. AE-SA achieves the highest AUC in the list of all models.

Model	Accuracy	Micro-F1	Aspect Size
ABAE	0.349	0.349	37
ABAE_HP	0.411	0.411	37
ABAE_FT	0.266	0.266	37
ABAE_FT_HP	0.383	0.383	37
ABAE_SH	0.325	0.325	40
AE-SA	0.350	0.350	40
AE-CSA	0.341	0.341	50

Tab. 7.3.: Highest accuracy, highest micro F1 score and corresponding aspect size for all models for the task of aspect extraction on Sentihood dataset. ABAE_FT denotes ABAE model initialized with pre-trained GloVe embeddings finetuned to Sentihood dataset. ABAE_HP and ABAE_FT_HP are the models ABAE and ABAE_FT with low precision classes removed.

7.1.1 Transfer Learning

7.1.1.1. ABAE

The ABAE experiments were conducted with varying experimental setups. The results are shown in table 7.3. The highest accuracy is obtained for 37 aspects when the ABAE model is initialized with pre-trained GloVe embeddings trained on Wikipedia and Gigaword corpus.

1. **ABAE experiments with pre-trained GloVe embeddings** (He et al., 2017) had performed their experiments on restaurant and laptop review datasets and had obtained the highest number of coherent clusters on the restaurant dataset. The dataset was annotated with 6 gold standard aspects, and the experiments were conducted for 14 inferred aspects. To obtain the maximum number of coherent clusters, we adopted the ratio of aspects and labels for our experiments. As an initial heuristic approach, we conducted experiments for aspect sizes 14, 28, 30, 37, 38, 39, and 42. Subsequently, the experiments were performed over an aspect count of 10 to 100. A cluster mapping is manual and becomes increasingly difficult with a growing number of aspects, mapping was done only for 10 to 50 aspects.

   Quantitative analysis of inferred aspects: The ABAE model obtains the highest coherence score for aspect 10 when measured by the UMass coherence

score. When measured by WETC, the highest coherence is obtained at aspect size 40.

Classification Accuracy The highest micro F1 score was achieved for aspect size 37. As shown in the table 7.3, this is the highest micro F1 score obtained amongst all experiments without omitting any class labels. Removal of classes that were having a null F1 score in consecutive experiments, improved the micro F1 score to 0.411. The initial experiments with the ABAE model were performed with Word2Vec embeddings trained on the Sentihood dataset. Model training with GloVe pre-trained embeddings increased the classification accuracy and micro F1 score by 54.26% from the Word2Vec experiment. This motivated the use of Glove embeddings for all subsequent experiments.

2. **ABAE experiment post removal of gibberish data** Removal of nouns, three-lettered, and less than three-lettered words from the training corpus increased 6 coherent inferred aspects, but it did not improve the classification accuracy. As the experiment was conducted for one aspect size, we did not calculate the coherence score for this experiment, as a single coherence score cannot be used to compare the experiment or model with the rest.

3. **ABAE experiments with finetuned GloVe embeddings**

   The micro F1 score obtained was 0.266 which was lower than the highest micro F1 score obtained for the ABAE model (aspect size 37). So we limited the number of experiments to one aspect size. Removal of classes that were having a null F1 score in consecutive experiments, improved the micro F1 score to 0.383.

Qualitative analysis of inferred aspects: We compare the model's understanding of topic clusters with human understanding by manually studying the aspects generated from our experiments. We also provide a record of our findings in table 7.4. In the table, we provide a count of coherent and relevant, coherent but irrelevant and incoherent clusters. For aspect count 10, a manual study of the inferred aspects reveal that only 3 clusters are coherent. Amongst the 3 clusters, only 2 can be assigned a gold standard label, and the third is mapped to 'general'. An example is shown below:

Coherent cluster which is mapped to 'general' label

"outskirt|0.58713293 suburb|0.5857716 bayswater|0.55934393 adjoining|0.5543636 situated|0.53446364 picturesque|0.5337991 neighbourhood|0.5108886 rosedale|0.50137115 overlooking|0.4937445 wooded|0.49081147"

		Coherent Clusters		
Model	Aspect Count	Relevant	Not Relevant	Incoherent Clusters
AE-CSA	30	6	11	13
AE-SA	30	4	7	19
ABAE_SH	30	7	2	21
ABAE	30	6	7	17
NMF	30	2	0	28
K-means	30	3	2	25

Tab. 7.4.: Qualitative analysis of inferred aspects: number of coherent and relevant clusters, number of coherent and not relevant clusters, number of incoherent clusters obtained from experiments for aspect size 30. A cluster is relevant if its topic is related to Sentihood dataset.

The 'general' aspect label is not only synonymous with incoherent clusters, but also to clusters which are coherent and relevant but do not have a suitable Sentihood gold label to associate with. The 7 incoherent clusters in the experiment included **a)** cluster of misspelled words like *'blimey', 'tellin', 'prob', 'aint'*, **b)** cluster of words belonging to different parts of speech in the English language, such as nouns, conjunctions, adverbs and adjectives, **c)** heterogeneous clusters that do not convey any topic, **d)** mixed clusters which convey more than one topic. In the ABAE experiment initialized with finetuned embeddings, we observe inferred aspects with mixed topics. Although they did not have a direct label for the best association, we used human inference to map the topics to the gold labels that were semantically related to the topic. As an example when a single aspect included words from varying topics like *'science', 'natural', 'museum and olympics', 'football', 'liverpool'*, the inferred aspect was mapped to the label 'touristy'. In comparison to the inferred aspects obtained from pre-trained GloVe embeddings used in our earlier experiments, the inferred aspects from finetuned GloVe embeddings are more related to the context of Sentihood and easier to be assigned in the cluster maps. From an overall perspective, we observed that some clusters were more difficult to interpret than others. As shown in the first two examples given below, the inferred aspects can be easily mapped to the respective labels 'price' and 'multicultural'. But the third example shown below that consists of alphanumerical terms, is difficult to be associated with a gold label. It looks incoherent, but a human observer will have to use general domain knowledge to filter terms like 'n14'[1]. The term is highly likely to refer to a

[1]The N14 postcode district is a postal district within the N London - N postcode area. Information taken from `https://www.streetlist.co.uk/n/n14`

postcode in the city of London, which is the domain of the dataset, and so the cluster is mapped to the gold label 'transit-location'.

Price

" million |0.6206217 compared|0.5245044 per|0.52381444 total|0.49247202 income|0.47724462 excluding|0.46497005 sq|0.46454373 population|0.4597404 average|0.45596352 worth |0.436171 share|0.42922455 respectively|0.41132182 increase|0.4070933 increased|0.3890413 rate |0.38882133 roughly|0.3796498 housing|0.36251682 basis|0.36064082 lowest|0.3606255"

Multicultural

"ortega|0.50206417 colombian |0.42718196 barrrre|0.3930954 da|0.38973352 barrio|0.38580525 spanish |0.37710303 brazilian |0.37064406 gabriel|0.36198694 esp|0.35027727 angel|0.335748 newell|0.32469124 pietra|0.3143616 la|0.31377035 villa|0.304713 portuguese|0.29728857 spain|"

Transit-Location

lhr|0.560034 cck|0.52444386 lmc|0.47275835 onr|0.44274828 n14 |0.43470043 e16|0.42345968 w9|0.41541898 e13|0.41305134 x26|0.41184118 lsb|0.41094923 chk|0.4064253 cr2|0.4004686 swt|0.3958126 sm1|0.39084816 n12|0.37512136 btw|0.3726501 lk|0.37023437 etc|0.36542434 ect|0.35965353 n22|0.3560523 init|0.35558406 kno|0.34897137 se5|0.34291512 '

Hence human knowledge about the domain of the experimental dataset was important to prevent the loss of information in the experiments.

It was also observed from our experiments that as we increase the number of inferred aspects, the homogeneity of the clusters improves, and parallelly the interpretability of the aspects increase.

Remarks The objective of the ABAE model was to discover coherent aspects from the Sentihood dataset. The ensuing experiments unveiled some limitations of the dataset. As data preprocessing depends on the type and domain of the dataset, we could not filter nouns from the corpus, as this causes the removal of words depicting nationalities like 'Iranian' or 'Bangladeshi'. We wanted to preserve these words as they associate with the gold label 'multicultural'. In a similar context, we preserved the alphanumerical terms which are generally filtered in preprocessing pipelines. Table 7.5 shows the number of words that were represented in each model. Social platforms have a specified limit for user reviews. Hence the sentences are short, and preserving sentence context after preprocessing is difficult. The classification accuracy and micro F1 score observed in the experiments are lower than the scores

obtained by (He et al., 2017) for the restaurant and laptop dataset. We assume the difficulty of the domain, and the small size of the training corpora to be related to the low micro F1 score. As mentioned by (He et al., 2017), topic extraction and classification was more difficult with the beer corpus than the restaurant corpus, as the beer corpus had a higher number of aspects. Still, the clusters obtained by ABAE_SH with only 829 word representations, shows the importance of domain knowledge in the task of aspect extraction.

Model	Number of word embeddings
ABAE_SH	829
ABAE AE-SA AE-CSA	3501
ABAE_FT	3666

Tab. 7.5.: Number of words represented by word embeddings in each model. ABAE_FT is ABAE model initialized with finetuned pre-trained GloVe embeddings.

7.1.1.2. AE-SA

The experiments for the AE-SA model were performed for aspect count 10 to 100, and cluster mapping was performed for aspect count 10 to 50 to evaluate the classification performance of the model. All experiments were performed using GloVe embeddings pre-trained on Wikipedia dataset. Interpretability of inferred aspects was observed to be similar to the corresponding experiments with the ABAE model.

Quantitative analysis of inferred aspects: When measured using cluster coherence metrics, the AE-SA model was observed to be more coherent than the ABAE model. The model obtained the highest coherence score when measured with the WETC metric.

Classification Accuracy The training loss of the AE-SA model achieves a significant reduction compared to ABAE, but the micro F1 score and classification accuracy do not vary for experiments with 10 to 50 aspects. As observed in table 7.3, the best Micro F1 score was obtained for inferred aspect count 40 and is approximately equal to the best accuracy achieved for ABAE model.

Qualitative analysis of inferred aspects: The inferred aspects consist of coherent and incoherent aspects such as some coherent aspects are relevant, and some are

irrelevant to the Sentihood gold standard aspect labels. Similar to the experiments with ABAE model, the inferred aspects include news and media, adverbs, adjectives, district and municipality, natural terrains, health and medicine, family relations, and sports. Amongst incoherent we observe clusters of misspelled words and mixed topic clusters.

Remarks In the experiments with the ABAE model, we observed that most inferred aspects were coherent but not relevant to the domain of Sentihood. The aspect clusters were more influenced by the distributional similarity of the words in the GloVe embedding space, than the context of the words in the dataset. Notably, the micro F1 score obtained by using GloVe embeddings pre-trained on Sentihood dataset was higher than the scores obtained for pre-trained GloVe embeddings fine-tuned on Sentihood corpus, as well as GloVe embeddings trained on the Sentihood corpus. Also, the coherence metric measures showed that ABAE_SH obtains the most coherent clusters, manual observation of the clusters revealed the two major problems: **a)** words were not being clustered according to their global context in the input document **b)** implicit aspects such as *'live', 'quiet'* had very low or null precision and recall scores in the classification report.

(Luo et al., 2019) interpreted this as a lack of knowledge about the correct word sense used in the sentence. They addressed the problem of word polysemy with a hierarchical sememe attention layer. As shown in the tables 7.1 and 7.2, the AE-SA model generates clusters that are more coherent than ABAE. The micro F1 score and classification accuracy of the model was lower than ABAE. We consider a set of reasons to explain the results.

- The sentences in Sentihood are short in length, and opinion targets or aspect terms are majorly nouns or alphanumerics. Nouns and alphanumericals do not belong to English language grammar, and hence their senses and sememes were not available on WordNet. Correspondingly, they were represented as a null vector in the sentence embedding.

- In WordNet, each sense of a word is represented as a synset with an associated part of speech. (Luo et al., 2019) aggregated a fixed count of senses and sememes for each word. In our experiments, we aggregated all the sememes of a sense, and all the word senses to represent the word. This may lead to a stronger representation of words with a higher number of synonyms, compared to words with a smaller number of synonyms or no synonyms.

7.1.1.3. AE-CSA

(Luo et al., 2019) proposed a second model AE-CSA which adds an RNN structure to the AE-SA model. The RNN is added to help the sememe attention layer to learn the sequential relations between the words in a review sentence, and attain lexical-semantic information to interpret the correct word meaning in the global context.

Quantitative analysis of inferred aspects: The AE-CSA model obtains higher cluster coherence than ABAE for the UMass coherence metric but is lower than ABAE when measured with the WETC metric. In both the measurements, AE-SA attains a higher coherence score than AE-CSA.

Classification Accuracy The reduction of reconstruction loss is smaller for AE-CSA model. The classification accuracy stays the same as AE-SA model.

Qualitative analysis of inferred aspects: The Umass and WETC coherence metric scores the AE-SA model to be more coherent in comparison to the AE-CSA model, but a detailed comparison of individual aspect counts show that the number of incoherent aspects is reduced in the AE-CSA model. (7.4) Though we assume the presence of human bias in deciding the relevance of an inferred aspect, the AE-CSA obtains a higher frequency of comprehensible inferred aspects. Under the circumstances, it was also observed that cluster mapping was easier for this model.

Remarks The experiment shows that the addition of an RNN structure does not necessarily improve the coherence of the autoencoder based topic model. As a second remark, the improved interpretability of the clusters shows an improved understanding of the sequential relations between the words in the sentence.

7.1.2 Deep Learning

1. **ABAE experiments with Word2Vec embeddings** Initial experiments were conducted using Word2Vec embeddings trained on the Sentihood dataset. But in subsequent experiments, when the model was initialized with pre-trained GloVe embeddings, a significant improvement in micro F1 score and classification accuracy was obtained. This motivated the use of GloVe embeddings for all the future experiments in our thesis. As the results were not significant for the thesis, we did not perform qualitative and quantitative analysis for this experiment.

2. **ABAE_SH experiments with GloVe embeddings trained on Sentihood corpus**

   Quantitative analysis of inferred aspects: The ABAE_SH model obtains the highest cluster coherence as measured using the UMass coherence metric 7.1.

   The model achieves a micro F1 score 0.325 which is lower than the best micro F1 obtained with ABAE.

   Qualitative analysis of inferred aspects: The inferred aspects majorly represented the domain of the dataset. Most of the aspects were incoherent as they consisted of words that are semantically related only in the context of Sentihood, but not in the context of common human knowledge. Hence the manual interpretation of the aspects is difficult if the observer does not have prior information about the domain. Hence although ABAE_SH generates a higher number of incoherent clusters than ABAE, the process of manual cluster mapping was easier for this model.

   Remarks Word embeddings are used to represent each word as a dense vector of real numbers so that semantically related words have a higher cosine similarity than a pair of unrelated words. As ABAE_SH is initialized with GloVe embeddings trained on the Sentihood corpus, word pairs co-occurring in the corpus are highly likely to appear in the same cluster. Hence the aspect clusters substantially represent the domain information of the dataset, unlike embeddings trained on neutral knowledge corpora like Wikipedia. The model achieves a high coherence score when measured using the UMass coherence score that corresponds well with human coherence judgments (Mimno et al., 2011). A low score in terms of the WETC metric can be attributed to the fact that the WETC score is calculated using word representations trained on Wikipedia and Gigaword corpora.

7.1.3 Baseline Models

The experiments were performed for aspect sizes 10 to 100. As manual cluster mapping is difficult we cluster mapped the inferred aspects for aspect size 10 to 50 and obtained the classification results for the same. As both the baseline models achieve similar classification results, that is all sentences in the evaluation dataset get classified as *'general'*, we limit the discussion to quantitative and qualitative analysis of the models.

1. **NMF**

   Quantitative analysis of inferred aspects: The model obtains a higher cluster coherence than ABAE, AE-SA, and AE-CSA in terms of the UMass coherence metric.

   Qualitative analysis of inferred aspects: In all experimental results, we observed a mix of coherent and incoherent aspects. But unlike ABAE, ABAE_SH, AE-SA, and AE-CSA, no coherent aspect was found to be irrelevant to the domain of Sentihood. (7.4)

2. **K-means:**

   Quantitative analysis of inferred aspects: The K-means model achieves the lowest cluster coherence score in terms of UMass coherence and WETC metric.

   Qualitative analysis of inferred aspects: For K-means, the aspect embeddings are the centroids initialized on the embedding matrix for the Sentihood training corpus. The inferred aspect clusters were mostly incoherent.

Remarks In the K-means model, as we do not train any neural network unlike ABAE, the inferred aspect clusters were mostly incoherent and did not represent the topic distribution in the training corpus. This shows that only embedding based topic detection is incorrect or not useful, and additionally shows the impact of the autoencoder architecture to learn topic sentence distribution.

7.2 Discussion

The experiments in this thesis were based on the initial hypothesis that deep learning architectures that performed distinctly on product review datasets would achieve comparable performance on the Sentihood dataset. Our hypothesis was deduced on the knowledge that review sentences are texts written informally to share information with the community. Therefore they share similar language construct, have similar patterns of noise (misspelled words), and were subjected to similar data cleaning pipelines. The findings in this thesis were found to be in contrast to our initial hypothesis.

Class	Number of Samples
Price	500
Shopping	143
Transit-location	428
Nightlife	158
General	1180
Live	221
Safety	352
Multicultural	123
Green-nature	95
Touristy	49
Quiet	54
Dining	93

Tab. 7.6.: Number of training samples for each class in the Sentihood dataset

Dataset	Reviews	Labelled Sentences
Sentihood	5215	3529
Restaurant	52574	3400
Beer	1,586,259	9245

Tab. 7.7.: Statistics of Sentihood dataset (used in our experiments), Restaurant and Beer dataset used by (He et al., 2017)

We applied the ABAE model on the Sentihood dataset keeping the dataset preprocessing and experiment hyperparameters similar to the experimental environment used in the research paper. The count of coherent topics generated by the encoder-decoder module and the classification scores obtained post manual cluster mapping were observed to be low compared to the results achieved by (He et al., 2017). We can associate this with some limitations faced in our experiments:

- Difference in dataset statistics. The training corpus for our model consisted of 3723 data samples compared to the restaurant and beer dataset used by (He et al., 2017) that have 49,174 and 1,577,014 training samples respectively.

- Number of gold standard aspect classes in our dataset Sentihood is twice the number of classes in the Restaurant and Beer dataset used in the paper. Also, our dataset has 12 classes, which reduces the number of training samples for each class. (7.6)

- The dataset is imbalanced, with 'general' as the majority class, as shown in figure 7.1.

- The review sentences are short in length, and dataset preprocessing is constrained due to the domain of the dataset. The Sentihood dataset is based on question answer pairs about urban neighborhoods in a city. Each sentence includes target location entities that are masked using nouns, other nouns visible in the dataset are location names that convey information about the masked entities, names of people, daily life commodities like food and household objects, sports, shops and restaurants. An omission of nouns will reduce the length of the sentences, which can decrease the model's understanding of sentence context. Hence, nouns, adverbs, adjectives, words with less than 4 alphabets, alphanumericals were not removed to preserve domain context in sentences. As the sentences are short, we did not remove misspelled words for the same reason.

- Several inferred aspects that were relevant to the domain, could not be mapped to a gold standard aspect label, as the predefined list is not conclusive of all the aspects in the dataset. A detailed list of inferred aspects is given in section A.2. When pre-trained Glove embeddings are used, we get very coherent clusters on the topic of sports, city administration, poverty, household articles, city outskirts etc. A human interpreter can associate topics like city administration, city outskirts with the dataset, but due to lack of aspect labels, these informative topics were mapped as 'general'.

In the experiments with pre-trained embeddings, as the representative words for aspect embeddings are selected from the GloVe embedding space trained on neutral data sources like Wikipedia and Gigaword, the inferred aspects are heavily influenced by word semantics in the pre-trained embedding matrix and reflect a lack of understanding of the sentence and domain context. Still, the ABAE model initialized with pre-trained GloVe embeddings demonstrated the maximum classification accuracy on the Sentihood dataset. We applied finetuning of GloVe on our dataset, and also experimented with GloVe vectors trained on the Sentihood dataset. Initializing the embedding matrix with finetuned GloVe embeddings reduced the training loss but also decreased the classification accuracy. Experiments with GloVe trained on the Sentihood data also demonstrated a reduction in classification accuracy. To improve classification accuracy, we removed the classes with zero F1 scores in most experiments. For the ABAE model with pre-trained embeddings, this improved the micro F1 score by 18%. For finetuned embeddings, removing low precision classes improved the micro F1 score by 44%. This may suggest that more training data per class can improve the classification scores for our dataset. Still, in comparison, pre-trained embeddings demonstrated higher classification scores than finetuned embeddings. We assume that the classification accuracy can be impacted by several factors such as lack of data, imbalanced dataset, absence of supervision, and presence of human decision bias.

In all our experiments, we faced the fact that identification of aspect terms is very important for identifying the aspect category. Implicit aspects are not associated with a clear aspect term in sentences, and are difficult to understand even for humans. The evaluation of the experiments in this thesis involves a human interpreter, who resolves each inferred aspect to a gold aspect label. This introduces bias and dependency upon the domain knowledge and sentence understanding of the human interpreter. As aspects in the evaluation dataset were manually labeled (Saeidi et al., 2016), the classification results may also be influenced by the knowledge gap between the annotator and interpreter. As sentences are short in length (70 characters), this may also impact the ability of the attention-based encoder to learn the topic distribution in our training corpus.

As evaluated using UMass and WETC coherence scores, we find that ABAE_SH is the most coherent model when measured by the UMass metric. This may be because the distributional similarity of word embeddings are related to the context of the dataset they are trained on. As the UMass metric corresponds to human understanding, this shows that the deep learning model can learn more effective features from word embeddings trained on the dataset, compared to embeddings trained on neutral or non-domain data.

The AE-SA model is the most coherent model when measured by WETC. The AE-SA model is also more coherent than ABAE in terms of the Umass coherence score. This shows that attention-based selection of word sense and respective sememes improves the representation of word embeddings, and resultantly improves the model's ability to understand word and sentence semantics. The experiments also depict the advantage of the sememe attention layer in the AE-SA and AE-CSA models.

In the experiments with the AE-CSA model, we observed an increase in the number of coherent clusters. Table 7.4 is an example of coherent and relevant, coherent and irrelevant, and incoherent clusters for aspect count 30. As we cluster mapped for aspects 10 to 50, we selected 30 as a mean aspect size and an example for comparison in our report. We also observed an improvement in topic interpretability as manual cluster mapping was easier for the AE-CSA model compared to the rest of the experiments. This shows that word sequence information is important for aspect term extraction and aspect categorization.

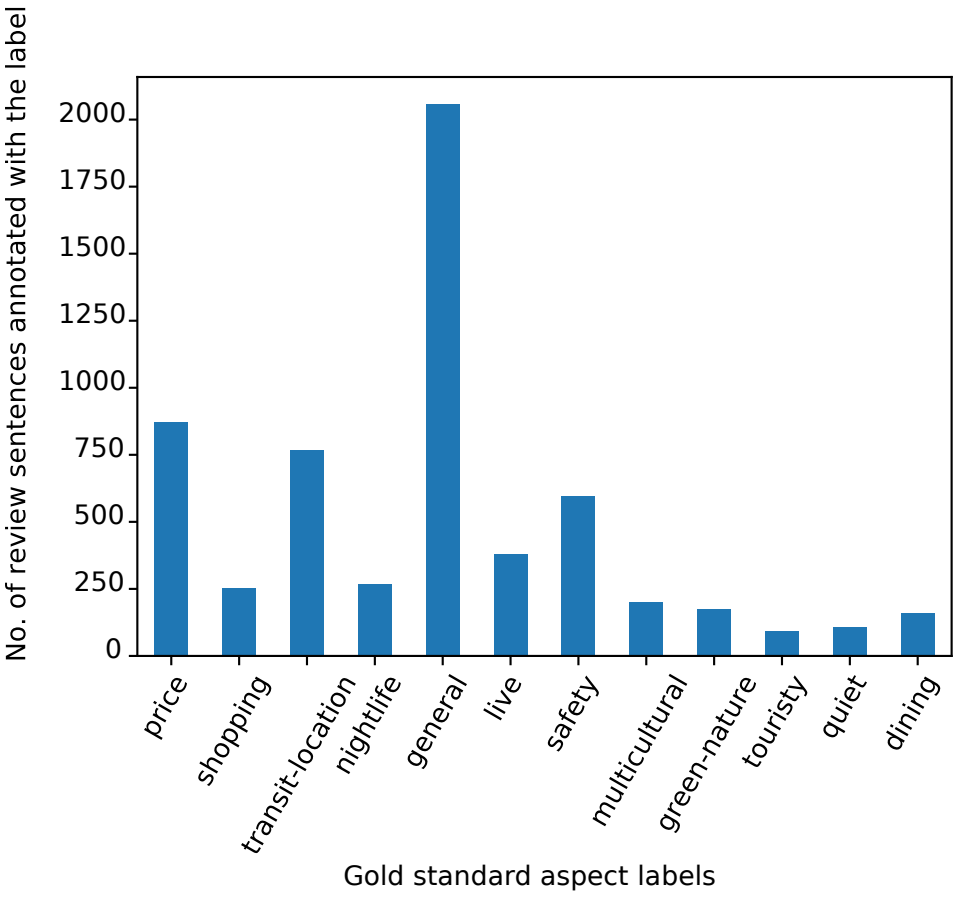


Fig. 7.1.: Distribution of gold standard aspect labels in the Sentihood dataset

8 Conclusion

In this thesis, we demonstrated the application of three unsupervised deep learning models for the task of aspect extraction on the Sentihood dataset. Our motivation was to perform an empirical study to understand if the state of the art unsupervised aspect extraction models performs well on our dataset. Our problem statement is inclusive of multiple challenges, given as:

- Unsupervised learning
- Short text
- Text was inclusive of multiple aspects that include explicit aspects (price, multicultural, shopping) and implicit aspects (live, quiet).
- Influence of domain constraints on data preprocessing pipeline
- Lack of training data

We performed experiments with pre-trained GloVe embeddings trained on a larger but neutral corpus, pre-trained embeddings finetuned to our dataset, and embeddings trained directly on our dataset. We measured the coherence of the inferred aspects using Coherence metrics UMass and WETC, and calculated classification scores. From our results, we can list a few inferences given as:

- Classification scores on product reviews are not a conclusive baseline for the task of aspect extraction.
- As aspects in text is strongly related to the domain of the text (Da'u and Salim, 2019), it is also useful to choose an approach based on the domain of the dataset.
- When selecting an unsupervised approach for performing aspect extraction on a given dataset, it is important to compare the aspect labels of both the datasets.

- Implicit aspects are harder to extract than explicit aspects.

- When classification scores are compared between two experiments on varying datasets, it is important to note whether the best scores were obtained on explicit aspect extraction only, or also included implicit aspects. An aspect extraction method for conversational text should be selected based on the latter's aspect extraction performance over explicit as well as implicit aspects. This is because unlike product reviews, conversational text may not have explicitly available aspect terms.

Appendix A

A.1 List of English stopwords in the NLTK toolkit

['i', 'me', 'my', 'myself', 'we', 'our', 'ours', 'ourselves', 'you', "you're", "you've", "you'll", "you'd", 'your', 'yours', 'yourself', 'yourselves', 'he', 'him', 'his', 'himself', 'she', "she's", 'her', 'hers', 'herself', 'it', "it's", 'its', 'itself', 'they', 'them', 'their', 'theirs', 'themselves', 'what', 'which', 'who', 'whom', 'this', 'that', "that'll", 'these', 'those', 'am', 'is', 'are', 'was', 'were', 'be', 'been', 'being', 'have', 'has', 'had', 'having', 'do', 'does', 'did', 'doing', 'a', 'an', 'the', 'and', 'but', 'if', 'or', 'because', 'as', 'until', 'while', 'of', 'at', 'by', 'for', 'with', 'about', 'against', 'between', 'into', 'through', 'during', 'before', 'after', 'above', 'below', 'to', 'from', 'up', 'down', 'in', 'out', 'on', 'off', 'over', 'under', 'again', 'further', 'then', 'once', 'here', 'there', 'when', 'where', 'why', 'how', 'all', 'any', 'both', 'each', 'few', 'more', 'most', 'other', 'some', 'such', 'no', 'nor', 'not', 'only', 'own', 'same', 'so', 'than', 'too', 'very', 's', 't', 'can', 'will', 'just', 'don', "don't", 'should', "should've", 'now', 'd', 'll', 'm', 'o', 're', 've', 'y', 'ain', 'aren', "aren't", 'couldn', "couldn't", 'didn', "didn't", 'doesn', "doesn't", 'hadn', "hadn't", 'hasn', "hasn't", 'haven', "haven't", 'isn', "isn't", 'ma', 'mightn', "mightn't", 'mustn', "mustn't", 'needn', "needn't", 'shan', "shan't", 'shouldn', "shouldn't", 'wasn', "wasn't", 'weren', "weren't", 'won', "won't", 'wouldn', "wouldn't"]

A.2 Coherent clusters that were mapped to 'General' aspect label

During the process of cluster mapping we observe coherent clusters with indicative words which don't contradict the domain of Sentihood, but which don't clearly align with available gold aspects. These clusters can also be claimed as ambiguous. For example time and seasons such as *'December','season','spring'*. A list of inferred aspect clusters that are related to Sentihood but could not be mapped to available gold aspect labels are:

Location names:
hampstead|0.5944066 essex|0.5869026 brighton|0.58188134 surrey|0.57932633 bexley|0.5754813 hertfordshire|0.5630959 sussex|0.5510436 cheshire|0.54263794

yorkshire|0.5243926 lancashire|0.52310246 salisbury|0.51557237 pancras|0.51534176 chelmsford|0.51423824 bayswater|0.5066268 stockport|0.5043479 lambeth|0.5000826 chiswick|0.4962823 guildford|0.4960363 windsor|0.4947791 ruislip|0.48552603 altrincham|0.48346144 borough|0.47641924 byfleet|0.47494882 bromley|0.47420067 albans|0.47219557 brook|0.46204886 rotherhithe

Adverbs:
frankly|0.63444495 awful|0.6211878 downright|0.6191932 incredibly|0.6186258 scary|0.59934634 terribly|0.59301347 extensively|0.39722884 reluctantly|0.3940806 defiantly|0.3924225

Adjectives:
stunning|0.39149374 score|0.30027223 impressive|0.29937053 10min|0.29686284 midway|0.26927602 quarter|0.2653371 record|0.2651354 berth|0.26355973 advantage|0.26306129 turistic|0.2557622 dodgyest|0.24978688 straight|0.24948844 broley|0.24719387 n9|0.24535023 lacklustre|0.23786575 game|0.23317732 winning|0.23311117 vote|0.23100662 respectable|0.22747856

Outskirts:
district|0.4075045 hamlet|0.38806248 village|0.3507074 boho|0.34488845 municipality|0.34403905 situated|0.34374645 shaftsbury|0.33902544 adjoining|0.32676104 neighbouring|0.32096007 haa|0.3095152 hte|0.29534265 bohemian|0.28946415 southeastern|0.28886026 predominantly|0.2790713 cemetary|0.27733788 hertfordshire|0.27547878 geographically|0.275148 county|0.27447385 vilagey|0.26701343 populated|0.26599672 westend|0.2634368 daventry|0.26119733 rural|0.26117098 rushbrooke|0.26081914 administratively|0.2602126

A.3 Coherent clusters that are not related to Sentihood dataset

A list of inferred aspect clusters that we see during the experiments and that were not relevant to Sentihood are :

Sports:
olympic|0.37176466 austria|0.36854628 downhill|0.36776963 skating|0.33752602 bt|0.32763338 beckie|0.32420927 berth|0.31567878 cycling|0.3121624 sweden|0.30405664 rd|0.30392408 nightjar|0.30031222 olympics|0.29438895 sixth|0.2889009 favourite|0.2872709 semi|0.2866521 spain|0.2843637 round|0.27815005 france|0.27341875 denmark|0.26918954

world|0.26675022 okayish|0.26638237 tennis|0.2635575

Media:
news|0.5088891 abc|0.49469417 radio|0.48866123 channel|0.48472577 daily|0.48060906 website|0.46782967 tv|0.45024738 independant|0.42876396 interview|0.39655772 standart|0.38140577 reported|0.37786114 network|0.3751349 biased|0.34756565 outlet|0.34014267 journalism|0.32321295 agency|0.3145572 article|0.31118572 globe|0.3019926 uni|0.30086014 fox|0.2992661 blacked|0.29575196 8pm|0.2954941 paper|0.29259044 mudlards|0.28564 printed|0.28557357 online|0.28413707 advertising|0.28243273 archive|0.2792227 info|0.27549684

Technology:
specification|0.58589804 html|0.46543515 wifi|0.44159678 standard|0.43524203 dial|0.43117756 hardware|0.42702797 interchange|0.4266458 code|0.42415446 alternatively|0.41963857 dual|0.41742057 defined|0.40901023 system|0.39976907 http|0.39445114 using|0.38865507 communication|0.379928 stack|0.37762502 criterion|0.37704837 basic|0.36794186 design|0.3650022 multiple|0.35986897 wiki|0.35848856

Health and medicine:
lung|0.5228738 skin|0.51102597 cause|0.50647604 acne|0.4960347 drug|0.4694546 ect|0.44995275 naturally|0.44601488 heart|0.4459312 patient|0.43465823 pain|0.4281281 arterial|0.4208013 exposed|0.40381506 risk|0.40356395 animal|0.38664314 extraction|0.38391766 condition|0.37597135 plague|0.37216753 organic|0.3719057 natural|0.36795545 drinking|0.3670035 caused|0.3639645 type|0.3636599 concentration|0.3626625 milk|0.35904145 pollution|0.3577642 suffers|0.356227 deprivation|0.35622454 excess|0.35611022 surface|0.35592026 cow|0.3547973 regeneration|0.35116965 pregnant|0.3497048

Bibliography

Ahmed, Hosameldin, M. Wong, and Asoke Nandi (2018). „Intelligent condition monitoring method for bearing faults from highly compressed measurements using sparse over-complete features“. In: *Mechanical Systems and Signal Processing* 99, pp. 459–477 (cit. on p. 10).

Angelidis, Stefanos and Mirella Lapata (2018). „Summarizing opinions: Aspect extraction meets sentiment prediction and they are both weakly supervised“. In: *arXiv preprint arXiv:1808.08858* (cit. on pp. 8, 50).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2014). „Neural machine translation by jointly learning to align and translate“. In: *arXiv preprint arXiv:1409.0473* (cit. on pp. 10, 11).

Belford, Mark, Brian Mac Namee, and Derek Greene (2018). „Stability of topic modeling via matrix factorization“. In: *Expert Systems with Applications* 91, pp. 159–169 (cit. on p. 36).

Bishop, Christopher M. (2006). *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag (cit. on p. 35).

Blei, David M, Andrew Y Ng, and Michael I Jordan (2003). „Latent dirichlet allocation“. In: *Journal of machine Learning research* 3.Jan, pp. 993–1022 (cit. on p. 11).

Bravo-Alcobendas, D. and Carlos Sorzano (2009). „Clustering of biomedical scientific papers“. In: pp. 205 –209 (cit. on p. 36).

Britz, Denny (2016). *Attention and Memory in Deep Learning and NLP*. en-US (cit. on p. 10).

Brun, Caroline and Vassilina Nikoulina (2018). „Aspect Based Sentiment Analysis into the Wild“. In: *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*. Brussels, Belgium: Association for Computational Linguistics, pp. 116–122 (cit. on p. 18).

Dai, Hongliang and Yangqiu Song (2019). „Neural aspect and opinion term extraction with mined rules as weak supervision“. In: *arXiv preprint arXiv:1907.03750* (cit. on pp. 3, 7).

Da'u, Aminu and Naomie Salim (2019). „Aspect extraction on user textual reviews using multi-channel convolutional neural network“. In: *PeerJ Computer Science* 5, e191 (cit. on pp. 7, 66).

Ding, Ran, Ramesh Nallapati, and Bing Xiang (2018). „Coherence-Aware Neural Topic Modeling“. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 830–836 (cit. on p. 49).

Dingwall, Nicholas and Christopher Potts (2018). „Mittens: an Extension of GloVe for Learning Domain-Specialized Representations“. In: *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*. New Orleans, Louisiana: Association for Computational Linguistics, pp. 212–217 (cit. on p. 44).

Dong, Li, Furu Wei, Chuanqi Tan, et al. (2014). „Adaptive Recursive Neural Network for Target-dependent Twitter Sentiment Classification“. In: *The 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. ACL (cit. on pp. 22, 23).

Dong, Zhendong, Qiang Dong, and Changling Hao (2010). „Hownet and its computation of meaning“. In: *Coling 2010: Demonstrations*, pp. 53–56 (cit. on p. 39).

Ganu, Gayatree, Noémie Elhadad, and A. Marian (2009). „Beyond the Stars: Improving Rating Predictions using Review Text Content“. In: *WebDB* (cit. on pp. 4, 19).

Goodfellow, Ian, Yoshua Bengio, and Aaron Courville (2016). *Deep Learning*. `http://www.deeplearningbook.org`. MIT Press (cit. on p. 9).

He, Ruidan, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier (2017). „An Unsupervised Neural Attention Model for Aspect Extraction“. In: *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vancouver, Canada: Association for Computational Linguistics, pp. 388–397 (cit. on pp. 4, 27–29, 38, 41–43, 45, 53, 57, 62).

Introduction to autoencoders. (2018). en (cit. on p. 9).

Jebbara, Soufian and Philipp Cimiano (2019). „Zero-Shot Cross-Lingual Opinion Target Extraction“. In: *arXiv preprint arXiv:1904.09122* (cit. on p. 7).

Jiang, Qingnan, Lei Chen, Ruifeng Xu, Xiang Ao, and Min Yang (2019). „A challenge dataset and effective models for aspect-based sentiment analysis“. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 6281–6286 (cit. on p. 21).

Jin, Huiming, Hao Zhu, Zhiyuan Liu, et al. (2018). „Incorporating chinese characters of words for lexical sememe prediction“. In: *arXiv preprint arXiv:1806.06349* (cit. on p. 30).

Jing, Li-Ping, Hou-Kuan Huang, and Hong-Bo Shi (2002). „Improved feature selection approach TFIDF in text mining“. In: *Proceedings. International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, pp. 944–946 (cit. on p. 11).

Karamanolakis, Giannis, Daniel Hsu, and Luis Gravano (2019). „Leveraging just a few keywords for fine-grained aspect detection through weakly supervised co-training“. In: *arXiv preprint arXiv:1909.00415* (cit. on p. 8).

Kim, Yoon (2014). „Convolutional neural networks for sentence classification“. In: *arXiv preprint arXiv:1408.5882* (cit. on p. 4).

Liao, Ming, Jing Li, Haisong Zhang, et al. (2019). „Coupling Global and Local Context for Unsupervised Aspect Extraction“. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 4571–4581 (cit. on p. 7).

Loper, Edward and Steven Bird (2002). „NLTK: The Natural Language Toolkit“. In: *Proceedings of the ACL-02 Workshop on Effective Tools and Methodologies for Teaching Natural Language Processing and Computational Linguistics - Volume 1*. ETMTNLP '02. Philadelphia, Pennsylvania: Association for Computational Linguistics, 63–70 (cit. on p. 37).

López, Dionis and Leticia Arco (2019). „Multi-domain Aspect Extraction Based on Deep and Lifelong Learning“. In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Ed. by Ingela Nyström, Yanio Hernández Heredia, and Vladimir Milián Núñez. Cham: Springer International Publishing, pp. 556–565 (cit. on p. 7).

Luo, Ling, Xiang Ao, Yan Song, et al. (2019). „Unsupervised Neural Aspect Extraction with Sememes“. In: *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*. International Joint Conferences on Artificial Intelligence Organization, pp. 5123–5129 (cit. on pp. 4, 30, 33, 58, 59).

Ma, Dehong, Sujian Li, Fangzhao Wu, Xing Xie, and Houfeng Wang (2019). „Exploring Sequence-to-Sequence Learning in Aspect Term Extraction“. In: *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*. Florence, Italy: Association for Computational Linguistics, pp. 3538–3547 (cit. on p. 7).

Ma, Yukun, Haiyun Peng, and Erik Cambria (2018). „Targeted Aspect-Based Sentiment Analysis via Embedding Commonsense Knowledge into an Attentive LSTM.“ In: *Aaai*, pp. 5876–5883 (cit. on p. 37).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge, UK: Cambridge University Press (cit. on pp. 14, 16).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean (2013a). „Distributed representations of words and phrases and their compositionality“. In: *Advances in neural information processing systems*, pp. 3111–3119 (cit. on p. 12).

Mikolov, Tomas, Kai Chen, Greg Corrado, and Jeffrey Dean (2013b). „Efficient estimation of word representations in vector space“. In: *arXiv preprint arXiv:1301.3781* (cit. on p. 12).

Miller, George A (1995). „WordNet: a lexical database for English“. In: *Communications of the ACM* 38.11, pp. 39–41 (cit. on p. 14).

Mimno, David, Hanna M. Wallach, Edmund Talley, Miriam Leenders, and Andrew McCallum (2011). „Optimizing Semantic Coherence in Topic Models“. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. EMNLP '11. Edinburgh, United Kingdom: Association for Computational Linguistics, 262–272 (cit. on pp. 49, 60).

Moon, T. K. (1996). „The expectation-maximization algorithm“. In: *IEEE Signal Processing Magazine* 13.6, pp. 47–60 (cit. on p. 35).

Pang, Bo and Lillian Lee (2008). „Opinion Mining and Sentiment Analysis“. In: *Found. Trends Inf. Retr.* 2.1–2, 1–135 (cit. on p. 3).

Parker, Robert and Linguistic Data Consortium (2011). *English gigaword fifth edition*. English. OCLC: 984977308 (cit. on p. 42).

Pedregosa, F., G. Varoquaux, A. Gramfort, et al. (2011). „Scikit-learn: Machine Learning in Python“. In: *Journal of Machine Learning Research* 12, pp. 2825–2830 (cit. on pp. 36, 40).

Pennington, Jeffrey, Richard Socher, and Christopher D. Manning (2014). „GloVe: Global Vectors for Word Representation“. In: *Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543 (cit. on pp. 13, 42, 46).

Pontiki, Maria, Dimitris Galanis, John Pavlopoulos, et al. (2014). „SemEval-2014 Task 4: Aspect Based Sentiment Analysis“. In: *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*. Dublin, Ireland: Association for Computational Linguistics, pp. 27–35 (cit. on pp. 3, 18).

Pontiki, Maria, Dimitris Galanis, Haris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos (2015). „SemEval-2015 Task 12: Aspect Based Sentiment Analysis“. In: *Proceedings of the 9th International Workshop on Semantic Evaluation (SemEval 2015)*. Denver, Colorado: Association for Computational Linguistics, pp. 486–495 (cit. on pp. 4, 19, 20).

Rajaraman, Anand and Jeffrey David Ullman (2011). „Data Mining“. In: *Mining of Massive Datasets*. Cambridge University Press, 1–17 (cit. on p. 11).

Saeidi, Marzieh, Guillaume Bouchard, Maria Liakata, and Sebastian Riedel (2016). „SentiHood: Targeted Aspect Based Sentiment Analysis Dataset for Urban Neighbourhoods“. In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. Osaka, Japan: The COLING 2016 Organizing Committee, pp. 1546–1556 (cit. on pp. 4, 23, 43, 47, 64).

Stenetorp, Pontus, Sampo Pyysalo, Goran Topić, et al. (2012). „BRAT: A Web-Based Tool for NLP-Assisted Text Annotation“. In: *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*. EACL '12. Avignon, France: Association for Computational Linguistics, 102–107 (cit. on pp. 18, 19).

Vijayarani, S, Ms J Ilamathi, and Ms Nithya (2015). „Preprocessing techniques for text mining-an overview“. In: *International Journal of Computer Science & Communication Networks* 5.1, pp. 7–16 (cit. on p. 11).

Wang, Bo and Min Liu (2015). „Deep learning for aspect-based sentiment analysis“. In: *Stanford University report* (cit. on p. 4).

Wang, Wenya and Sinno Jialin Pan (2018). „Transition-based Adversarial Network for Cross-lingual Aspect Extraction.“ In: *IJCAI*, pp. 4475–4481 (cit. on p. 7).

Wang, Zhaoxia, Chee Seng Chong, Landy Lan, et al. (2016). „Fine-grained sentiment analysis of social media with emotion sensing“. In: *2016 Future Technologies Conference (FTC)*. IEEE, pp. 1361–1364 (cit. on p. 3).

Zhao, Chao and Snigdha Chaturvedi (2019). „Weakly-Supervised Opinion Summarization by Leveraging External Information“. In: *arXiv preprint arXiv:1911.09844* (cit. on p. 8).

Zoph, Barret, Deniz Yuret, Jonathan May, and Kevin Knight (2016). „Transfer learning for low-resource neural machine translation“. In: *arXiv preprint arXiv:1604.02201* (cit. on p. 5).

List of Figures

List of Tables