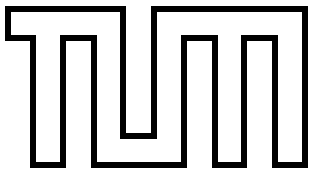


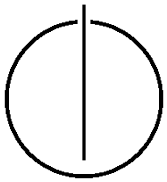
DEPARTMENT OF INFORMATICS

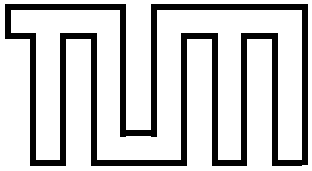
TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Data Engineering & Analytics

Multilingual Opinion Mining on Social Media Comments Using Unsupervised Neural Clustering Methods

Mainak Ghosh





DEPARTMENT OF INFORMATICS

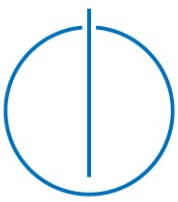
TECHNICAL UNIVERSITY OF MUNICH

Master's Thesis in Data Engineering & Analytics

Multilingual Opinion Mining on Social Media Comments Using Unsupervised Neural Clustering Methods

Mehrsprachiges Opinion Mining auf Kommentaren aus sozialen Medien unter Verwendung von unüberwachten neuronalen Clustering-Methoden

Author:	Mainak Ghosh
Supervisor:	PD Dr. Georg Groh
Advisor:	Gerhard Hagerer, M.Sc.
Submission Date:	October 15, 2019



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

Munich, October 15, 2019 Mainak Ghosh

Acknowledgement

First and foremost, I would like to express my deepest gratitude to my supervisor PD Dr. Georg Groh and my advisor Gerhard Hagerer for providing me this opportunity to conduct research and collaborate with them in their Social Computing Research Group. Their immense knowledge, enthusiasm, and motivation guided and enriched me all through my research. I couldn't imagine any better research lab to collaborate with.

I am deeply indebted to Gerhard Hagerer for providing interesting ideas, continuous advice, and insightful feedback in my research. His patience and encouragement helped me nourish my learning spirit, keeping me engrossed and fascinated with never a mundane moment. His congenial nature and easy accessibility played a vital role in timely completion of the thesis.

I would like to extend my sincere thanks to Hannah Danner for her invaluable insights into my research as an expert opinion researcher.

I would like to take this opportunity to appreciate all the members of the SocialROM group for the constructive discussion, helpful advice, and practical suggestions in our meetings.

Furthermore, I would like to extend my warm regards to my parents for their love and support. Finally, I would also convey my gratitude to my girlfriend, Shayoni for her unstinting support and encouragement, without whom it wouldn't have been possible to complete my thesis.

Abstract

Today, due to increasing popularity of social media such as social networking sites, online review sites, forums, blogs, news-articles etc., fine-grained opinion mining has become immensely beneficial to companies, policy makers, government to get human perception about their products. Automated aspect extraction is an effective way for studying products impacting policies, gaining insights about human-interest and culture. It is very likely that reviews are inclusive of multiple languages such as English, German, French, so opinion mining on multilingual data is a very important and challenging task in order to learn collective opinions.

In this master's thesis, we extended an unsupervised attention-based neural clustering model, proposed by he et al. (2017), for extracting coherent bilingual aspects in an effective manner. This thesis conducted in-depth analysis of different variants of word embedding techniques such as global vectors (GloVe) (Pennington et al., 2014), word2vec (Mikolov et al., 2013b), and fastText (Bojanowski et al., 2016) to understand the relation between the degree of granularity of coherent aspects and word embedding techniques. We also deployed multilingual word embeddings, MUSE (Conneau et al., 2017) for understanding the bilingual context. To the best of our knowledge, this is the first time when multilingual unsupervised and supervised embeddings (MUSE) is applied with *Attention-based Aspect Extraction* (ABAE) model (he et al., 2017).

We were able to evaluate the bilingual model by measuring the coherent aspects using different variants of coherence score calculation, thus creating a benchmark for multilingual ABAE with competitive results. This thesis also draws the aspect distribution over bilingual data which forms the basis for gaining insights about social opinion clusters.

Keywords: Opinion mining, Multilingual aspect extraction, Attention-based aspect extraction, Word embedding, MUSE, GloVe, Word2vec, FastText, Organic food, Semantic clustering, Social media analysis

Contents

List of acronyms

BoW Bag of Words 16

NLP Natural Language Processing 3

CNN Convolutional Neural Network 11

LSTM Long Short-term Memory

LDA Latent Dirichlet Allocation 5

ABAE Attention-based Aspect Extraction xi

SVM Support Vector Machine 9

CRF Conditional Random Fields 10

POS Part-of-Speech 10

GRU Gated Recurrent Unit 10

Bi-LSTM Bidirectional LSTM

CCA Canonical Correlation Analysis 21

CBOW Continuous Bag-of-Words Model

GloVe Global Vectors ix

LSA Latent Semantic Analysis 17

OOV Out of Vocabulary 19

MUSE Multilingual Unsupervised and Supervised Embeddings ix

SVD Singular Value Decomposition

NPMI Normalized Point-wise Mutual Information

GPU Graphics Processing Unit 36

JSON JavaScript Object Notation 40

URL Uniform Resource Locator

NLTK Natural Language Toolkit

1 Introduction

1.1 Motivation

Since the advent of web 2.0 and ubiquitous presence of internet, individuals exchange their ideas, opinions, and feelings in a broad spectrum (Mathapati et al., 2017). In this era of digitalization, online social media, blogs, forums, news-sites are omnipresent and an integral part of human life. People can now provide feedbacks, reviews about any product, service on companies' online websites, and social media any time (Arora et al., 2015). In fact, expressing concern or consent to government's policies is just a click of a button away. These opinions are valuable source of information for companies, organizations, and government to gain insights about peoples' thoughts. The insights gained are immensely beneficial to them to take the necessary decision towards the goal of achieving popularity, generating revenue. However, the humongous volume of information at an unprecedented velocity makes the scrutiny of these unstructured text impossible for humans; thus leading to the notion of automated opinion mining.

In order to gain insights from the opinionated text, machines need to understand the natural language, its complexity and subtle nuances, however, machine can only understand binary digits (0, 1). This initiates the need of machine, understanding human language. Hence it is an important and challenging task in the field of natural language processing (NLP). Recent advancement of deep learning in NLP tasks (Young et al., 2018), such as word representation (Mikolov et al., 2013b), attention mechanism in machine translation (Bahdanau et al., 2015), has proved to be promising for machine to understand human language, thus accelerating the automated opinion mining.

An opinion can be generated out of the reviews, even each sentence of a review expresses some opinions. However, as Liu (2012) stated, review level opinion and sentence level opinion do not necessarily specify individual's like or dislike. Below example, taken from Liu (2012),

"*although the service is not that great, I still love this restaurant*"

conveys a positive feeling, but it indicates negativity about service. Here, "*service*"

is referred as aspect term. Certainly aspect-based opinion is helpful for companies to understand opinion about different features of products, which is immensely beneficial to the industrial market as well as individuals can benefit from this. Hence aspect extraction is indeed an important task in opinion mining.

Aspect-based Opinion Mining

Aspect-based opinion mining is a key task in fine-grained opinion mining. The aspects of an entity in a sentence refer to the components and attributes of that entity. An aspect expression is the exact word, present in the sentence to represent the aspect of the entity (Liu, 2012). Liu (2012) stated that aspect can be explicit and implicit. An example about digital camera from Hu and Liu (2004) is given below to differentiate between explicit and implicit aspect:

"*The pictures are very clear. While light, it will not easily fit in pockets.*"

Here, "*pictures*" is explicitly mentioned in the text, so it is an explicit aspect. In the second sentence of the example, user is talking about size of the camera, though it's not mentioned in the text, so it is an implicit aspect. So implicit aspect is hard to be extracted compared to explicit aspect. Hu and Liu (2004) proposed a rule-based method to extract explicit aspects. They tried to find frequent set of words or phrases that occur together in some sentences using association mining[1]. However, this method is not robust in this task, since it might produce too many non aspect terms as well as inconvenient in finding less frequent aspect terms (Schouten and Frasincar, 2016). In due course, there were several models developed to extract implicit aspects (Su et al., 2008; Zhang and Zhu, 2013). Despite the hurdles, research communities have progressed in aspect-based opinion mining. Taking advantage of probabilistic graphical models (he et al., 2017) to devise supervised approaches for extracting aspects was one such endeavour.

Supervised approach however requires lot of labeled training data and faces domain adaptation issues (he et al., 2017). Even labeled data can be biased towards some opinions, thus making aspect extraction troublesome. Brody and Elhadad (2010) explained that unsupervised method is suitable for aspect extraction in online social media. Firstly, unsupervised method is able to work on large easily available voluminous dataset of opinionated texts unlike supervised method, which requires hardly obtainable annotated data, so unsupervised method does not experience the domain adaptability problem. Secondly, social media comments are usually

[1] Association mining, a field of data mining, tries to find out interesting correlation, frequent pattern, association or casual structure among the database items (Kumar and Chezian, 2012)

short, unstructured, in addition, may contain spelling and grammatical errors, some slang or specialized jargon (Brody and Elhadad, 2010), due to which supervised or rule-based methods or methods relying on lexicon, dictionaries do not work well. Unsupervised methods are not motivated by the lexical form, and can even take care of unknown words if they occur frequently enough in the dataset (Brody and Elhadad, 2010). Subsequently, topic modeling techniques, Latent Dirichlet Allocation (LDA) (Blei et al., 2002) and its variants, such as model proposed by Brody and Elhadad (2010), have become popular as unsupervised methods in aspect extraction (he et al., 2017). These topic modeling techniques even cluster the aspect terms into some meaningful categories unlike rule-based approach. For example, aspect terms "beef", "chicken", "pork", "potatoes" should lie in one cluster, which can then be labeled as food. In other words, this method determines the aspect using soft clustering of aspect terms (Lim and Buntine, 2016). Thus, generating coherent clusters of aspect terms, which determines entire aspect or category of the cluster, is a challenging task and worthy of exploration.

1.2 Problem Statement

Topic models in aspect extraction have been popular, since soft clustering inherently identifies aspects, without finding aspect terms and grouping them separately (he et al., 2017). Thereby, it is possible to understand different aspects over opinionated texts, which indeed helps to realize what people talk about on social media and online platforms. Industrial market is certainly interested in knowing this latent aspects or topics, since this knowledge helps market in visualizing discussion patterns. These insights even manipulate the business ideas towards revenue generation, such as accelerating excellence in online ads, recommendation etc. (Reisenbichler and Reutterer, 2019). Hence, categorizing a set of extracted aspect terms is an important task, as it leads to inferred aspect.

Latent Dirichlet Allocation (LDA) (Blei et al., 2002) based topic models and it's variants indeed have been fruitful in learning aspects from social media comments. However, he et al. (2017) mentioned that LDA-based model does not infer good quality of aspects, since cluster of aspect terms many times does not cater to words with similar meaning. Hence, lack of good coherent clustering of aspect terms leads to some biased aspect extraction of the comments. Having felt the need of more semantic similarity within cluster of aspect terms, he et al. (2017) came up with their model, Attention-based Aspect Extraction (ABAE).

ABAE certainly extracts better quality of aspects with respect to LDA-based model on monolingual dataset (he et al., 2017). However, people live in diverse world,

where mankind does not have common culture or common language. They express their opinions in their own languages and styles, thereby leading to wide variety of languages and cultures. In present age of globalization, one product is even available in multiple countries. Hence, the availability of products in global market makes the reviews available in multiple languages (Asnani and Pawar, 2017). This mandates extracting aspects from multilingual reviews, since it helps companies and organizations to know the opinion across the globe and understand the diversity in cultures and preferences; thus impacting their business decisions. This instigated us to extend ABAE model for multilingual opinion mining and investigate how the multilingual ABAE performs. Additionally, we analyzed if multilingual ABAE model generates coherent clusters of multilingual aspect terms as well as to what extent this multilingual ABAE model excels in generating good quality of aspects. This thesis also elaborates on the determination of optimal number of aspects, needed to get the insights about social media comments; hence, restricting general aspects.

1.3 Proposed Solution

This thesis, knowing the importance of multilingual opinion mining, proposes an ABAE-based model in order to gain insights from multilingual social media comments. Since our dataset were based on German and U.S. organic food, we were required to generate bilingual word embedding to categorize aspect of aspect terms out of our bilingual dataset. Aspect terms of a cluster should cohere with the inferred aspect. So we applied MUSE (Conneau et al., 2017) to align English word representation and German word representation in same word embedding space, such that semantically similar English and German words lie close to each other.

As a consequence, English and German word representations were necessary. In recent years, there have been several advancement in representing words in vector space. Word2vec (Mikolov et al., 2013b), GloVe (Pennington et al., 2014), and fastText (Bojanowski et al., 2016) are popular in learning linguistic relations, semantic and syntactic similarity. This thesis explored different variants of these embedding techniques to investigate which embedding model and it's variant gives high quality of aspects on our both dataset, by avoiding inferring biased aspects; eventually resulting in good quality of English and German word representation. Thereafter, MUSE functioning on these monolingual word representations helped in obtaining bilingual word embeddings for generation of coherent aspect terms. Additionally, this thesis explored hyperparameter optimization of number of aspects for organic food data and illustrated how optimal number of aspects changes in accordance to embedding methods and their variants. Hence, it helps in restricting nonsensical general clusters of aspect terms.

The generation of coherent aspects also demands for a quantitative measurement to identify how word embedding methods and different aspect sizes[2] perform. Human evaluation however is not a feasible option due to the large volume of aspects terms, therefore automatic coherence evaluation metric cropped up. This explains why this thesis explores different variants of automatic coherence evaluation metric, such as topic coherence score, proposed by Mimno et al. (2011) and word embedding cosine similarity (Wang et al., 2019; Ding et al., 2018). These were also beneficial to finding optimal number of aspect sizes. This thesis also analyses distribution of topics or categories over the bilingual dataset, which certainly aids differentiating the diverse culture, which was one of the motivation behind this thesis.

1.4 Outline

This section briefs about the orientation of this thesis and essays overview of each chapter.

Chapter 2

This chapter talks about recent scientific evolution and approaches regarding aspect extraction. It also portrays how this thesis differs from previous scientific achievements.

Chapter 3

This chapter focuses on all the necessary knowledge required to understand this thesis, for example, word vector representations, attention mechanism.

Chapter 4

This chapter talks about our base model ABAE as well as fine-tuning GloVe embedding model.

Chapter 5

We describe structure and statistics of our English and German organic food data in this chapter.

Chapter 6

This chapter elaborates the data preprocessing steps, all the experiments, which were performed for the task of aspect extractions over multilingual data and relevant results on our dataset. It also analyses the extent that ABAE is efficient and effective

[2]Aspect size is same as number of aspects

in the said purpose.

Chapter 7

This chapter summarizes our achievement throughout the thesis and addresses some research questions. It also elaborates some future scope of this thesis.

2 Related Work

This thesis mainly explores multilingual aspect extraction and hence in this chapter we outlined past and recent methodologies, endeavors, proposed by research communities in the context of aspect extraction and multilingual aspect extraction. This chapter also points out how this thesis differs from the state-of-the-art approaches.

2.1 Aspect Extraction

Advent of the World Wide Web and emergence of machine learning in natural language processing led to the evolution of opinion mining (Pang and Lee, 2008). Aspect extraction was not unattended to in this evolution; academics and research enthusiasts have expanded their knowledge in this sub-field since then. Schouten and Frasincar (2016) categorized aspect extraction techniques into five segments, which are frequency-based, syntax-based, supervised machine learning, unsupervised machine learning, and hybrid approaches.

In frequency-based approach, Hu and Liu (2004), based on the observation that aspects are generally surrounded by the opinion words, proposed a method to calculate the frequency of combinations of nouns, followed by application of association rules to prune those results (Schouten and Frasincar, 2016). Liu et al. (2005) and Hai et al. (2011) proposed similar kind of methods for aspect extraction. However, these methods lead to wrong consideration of high frequent non-aspect terms as aspect terms (Schouten and Frasincar, 2016).

Syntax-based approaches, on the other hand, find aspects by taking advantages of syntactic relations in sentences. Popescu and Etzioni (2005) and Qiu et al. (2011) proposed co-extraction of opinion words and aspect terms based on the syntactic relations (Li et al., 2018). These techniques are also able to extract less frequent aspect terms (Schouten and Frasincar, 2016). However, this concept is prone to erroneous results for informal social media comments, due to heavy dependence on syntactic patterns (Li et al., 2018).

In the continuous research endeavors with the variants architectures, Kessler and Nicolov (2009) came up with supervised machine learning model, support vector

machine (SVM)-based solution in order to understand which opinion expression is connected to which aspect term. However, domain adaptation is a big problem in supervised machine learning based model for aspect extraction. In order to address this shortcoming, Jakob and Gurevych (2010) proposed a method based on sequence labeling model, conditional random fields (CRF). CRF-based model was widely used for aspect extraction in supervised learning setting before deep learning gained popularity. For the first time, Poria et al. (2016) applied deep learning using word embeddings, enriched by part-of-speech (POS) tags for aspect extraction. This model outperformed all the then existing models (Huber and Spiliopoulou, 2019). Hence, current supervised state-of-the-art approaches are built upon deep neural network with word embeddings.

Supervised machine learning based methods however require lot of annotated data, which is expensive. This is the reason for which unsupervised machine learning based approaches have been popular for aspect extraction in academia and research communities. LDA-based model and its variants have been predominant as unsupervised models for the said task (Schouten and Frasincar, 2016). Apart from LDA, Schouten et al. (2018) and Dragoni et al. (2018) recently proposed unsupervised rule-based methods for aspect extraction (Huber and Spiliopoulou, 2019).

Unsupervised rule-based methods however fail to extract high level aspects, so Wu et al. (2018) proposed an effective hybrid unsupervised method, which combines rules and a deep gated recurrent unit (GRU).

2.2 Attention-based aspect extraction

Rule-based approaches heavily depend on predefined rules and do not categorize extracted aspect terms (he et al., 2017). On the other side, LDA-based approaches do not provide labeled aspect. Aspects are inferred from the clusters of aspect terms. Since LDA-based approaches use bag of words concept, extracted aspect terms are not required to be semantically coherent. This makes categorizing the clusters difficult (Schouten and Frasincar, 2016). In order to diminish these shortcomings, he et al. (2017) came up with an attention-based neural architecture, Attention-based Aspect Extraction (ABAE) model, following the fact that attention-based models had been effective in various natural language processing tasks, such as, machine translation (Bahdanau et al., 2015), sentence summarization (Rush et al., 2015). It conceptualizes the observation that all the words of a sentence are not relevant for that sentence, so attention mechanism pays more attention to some words, cohering with the sentence (Hu, 2018).

Similarly, Wang et al. (2017) proposed a model consisting of multiple attention networks on top of GRUs for extracting aspect terms and opinion terms. Zhu et al. (2018) proposed a model using bidirectional LSTM (Bi-LSTM) and convolutional neural network (CNN) for aspect-based text classification. They incorporated CNN-based attention mechanism for attention extraction over sentences and weighted aspect embedding for aspect extractions.

2.3 Multilingual Aspect Extraction

As compared to aspect extraction on English, aspect extraction on different languages is not so easy task. Learning linguistic structure and complexity are expensive tasks. Despite the facts, research communities have ventured into multilingual topic modeling and cross-lingual aspect extraction using parallel corpus[1], comparable corpus[2] and multilingual dictionaries (Hao and Paul, 2018). In this section, we elaborate researches in multilingual topic modeling, since topic modeling is a popular unsupervised technique for aspect extraction. Zhao (2006) and Zhao (2007) proposed bilingual topic models, which take advantage of word alignment and sentence-level parallel corpus. Furthermore, Zhu et al. (2013) proposed a model, based on LDA, for bilingual topic modeling using comparable corpora. However, parallel corpus and comparable corpus are expensive and hard to be obtained. There have been several model proposed, which take advantages of easily available multilingual dictionary (Boyd-Graber and Blei, 2009; Jagarlamudi and Daumé, 2010). In order to do more fine-grained topic modeling, such as understanding topics per segment of a document, Tamura and Sumita (2016) proposed bilingual segmented topic model. Apart from this, Asnani and Pawar (2016) incorporated semantic knowledge in LDA-based model for understanding topics in code-mixed[3] social media data. Huber and Spiliopoulou (2019) proposed an idea for multilingual aspect extraction using monolingual texts. They extracted aspects from monolingual texts and clustered them to multilingual topics.

In the context of cross-lingual opinion mining tasks, a model is generally trained on resource-rich source language; then the trained model is applied on target language with the help of parallel corpora and multilingual dictionary. Aspect-based task is not an exception either. CLOpinionMiner is a model for cross-lingual aspect extraction based on machine translation (Zhou et al., 2015). In this approach, annotated source dataset is translated to target language using machine translation. However, this method requires access to word alignment information, produced by machine

[1]Parallel corpus contains sentence-aligned documents in multiple languages
[2]Comparable corpus contains documents with similar themes in multiple languages
[3]Code-mixed data contains conversation between people, where they switch their language between multiple languages.

translation. In addition, machine translation is not effective for all languages. So to avoid the application of machine translation, Jebbara and Cimiano (2019) proposed a method, which leverages multilingual word embedding in CNN architecture for extracting aspects in target language. This method does not require any annotated data in target languages.

It has been observed so far that multilingual topic modeling is mostly based on LDA. Although, Huber and Spiliopoulou (2019) applied ABAE model in one of their experiments for extracting multilingual aspects, their goal was not generating coherent aspects but extracting aspect terms, and hence they filtered all the nouns having weight more than a threshold value as aspect terms. However, we did not change the proposed ABAE model. We also optimized the cluster size by observing the coherence score of the generated aspects. Huber and Spiliopoulou (2019) used algorithm proposed by Joulin et al. (2018) for multilingual word embedding, however, we used MUSE (Conneau et al., 2017) for that purpose. Prior to applying MUSE, this thesis explored several word embedding methods, such as GloVe (Pennington et al., 2014), word2vec (Mikolov et al., 2013b), fastText (Bojanowski et al., 2016), to identify which embedding excels in our monolingual dataset. Having decided that, this thesis explored MUSE on top of that embedding for multilingual aspect extraction and clustering. To the best of our knowledge, this is the first time when MUSE is applied with ABAE for multilingual aspect extraction. In these ways, this thesis differs from the existing approaches.

3

Theoretical Background

This chapter elucidates theoretical concepts, important for understanding the thesis.

3.1 Word Representations

Deep neural network is built upon mathematical model, which requires matrices and vectors for computation. Thereby, it is necessary for the network to accept numerical inputs. Hence words need to be represented as vectors, so that deep learning based architecture can also perform the computation over the non numerical dataset. One simple representation is one-hot encoding, which requires forming a vocabulary of distinct words in the given document. If the vocabulary represents N distinct words, where each word has an index to identify itself inside the vocabulary, then a word is represented in N dimensional vector form with the index of the word from the vocabulary set as 1 and remaining N-1 positions as 0. For example, consider a sentence:

"Apple is good for health, so I should buy apple."

Its vocabulary (ignoring the case) is represented as

{apple, is, good, for, health, so, i, should, buy}

so, word vector of "*apple*" is:

$$\text{w(apple)} = [1, 0, 0, 0, 0, 0, 0, 0, 0]$$

Similarly, word "*buy*" is represented as:

$$\text{w(buy)} = [0, 0, 0, 0, 0, 0, 0, 0, 1]$$

So, word embedding or word vector matrix is represented as:

$$\mathsf{W} = \begin{array}{c} \\ apple \\ is \\ good \\ . \\ buy \end{array} \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{pmatrix}$$

However, this matrix is heavily sparse. If a document has 100,000 unique words, then the corresponding word embedding size will be $100,000 \times 100,000$, which is very large but contains few non zero entries; thus leading to expensive computation and storage. This explains the need for dense and lower dimensional representation for words. Another aspect is that due to one-hot vector representation, all the words of the vocabulary are equidistant from each other in this vector space, but it is beneficial if semantically similar words lie close to each other. This accelerated the quest for another representation; leading to the advent of word2vec embedding consequently.

3.1.1 Word2vec

Mikolov et al. (2013b) came up with a groundbreaking word embedding, word2vec, which makes the semantically similar words lie closer in the lower dimensional embedding space (Yin and Shen, 2018). Thus this model learns high quality dense word vector from corpus. The authors presented two architectures of this embedding, skip-gram model and continuous bag-of-words model. Both are based on shallow neural network to generate meaningful representation of words. Due to much low computational cost, these models can generate word embeddings out of very large corpus; hence increasing efficiency in several order of magnitude than the then existing models. Word vectors also follow simple arithmetic operations and produce meaningful words . For example,

$$vector(\text{"}King\text{"}) - vector(\text{"}Man\text{"}) + vector(\text{"}Woman\text{"}) \approx vector(\text{"}Queen\text{"})$$

Skip-gram Model

Skip-gram model is trained to predict surrounding words within the defined context, given an input word. This context is defined by a window, which slides through the sentence. Given a context window size C, half the words of the sentence preceeding the input word and half the words following it, fall in the context window. These are called context words and the input word is called target word or focus word. For

example, consider a sentence,

"Berlin is the capital of Germany."

Having C = 3, target word and context words are exemplified below:

Context window (size = 3)	Target word	Context words
[Berlin is]	Berlin	is
[Berlin is the]	is	[Berlin the]
[is the capital]	the	[is capital]
.....	.	[..]
[of Germany]	Germany	of

Tab. 3.1.. This table shows that how context window is applied over sentences for context words and target word, which is essential for input and output layer in word2vec model.

Figure 3.1a shows the skip-gram model. To summarize the concept, this model tries to learn high quality target word vector, by maximizing the likelihood of words, $w_1, w_2, ..., w_C$ being in the context C with respect to target word x. Hence, the model minimizes the below loss function (Rong, 2014).

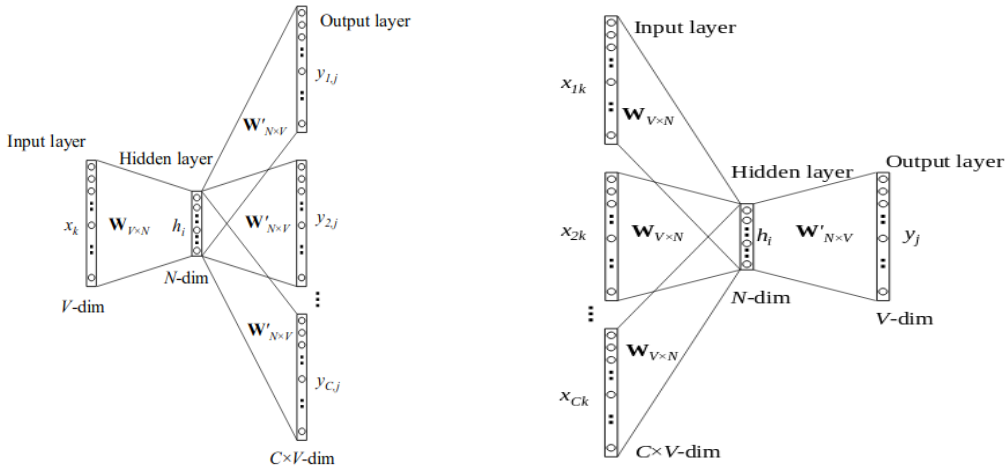
$$E = -logp(w_1, w_2, ..., w_C|x) \tag{3.1}$$

$$= -log \prod_{c=1}^{C} p(w_c|x) \tag{3.2}$$

where x and w_c are V-dimensional word vectors, $p(w_c|x)$ represents the probability that w_c matches the original c-th context word with respect to x. V represents the vocabulary size of the document. $y_{c,j}$ in the figure 3.1a represents the j-th unit of the word, w_c in the output layer. Although the window size is a hyperparameter, Mikolov et al. (2013b) noticed that increasing the window size leads to higher quality word vector, though it increases the computational complexity. Once the training is complete, learned weight matrix, W of size (V,N) is defined as word embedding matrix (see figure 3.1a). Each row of W represents dense and N-dimensional embedding for the corresponding word.

Continuous Bag-of-Words Model (CBOW)

This architecture is similar to the skip-gram model to a large extent. However, unlike skip-gram model, instead of predicting context words with respect to target word, CBOW model predicts the target word, given all the context words of the corresponding context window. Figure 3.1b depicts the CBOW model. Considering that $x_1, x_2, .., x_C$ are the context words within the context window of size C, $x_1, x_2, .., x_C$ are projected onto the hidden layer of N nodes (see figure 3.1b). Output of the hid-



(a) Skip-gram **(b)** CBOW

Fig. 3.1.. (Source: Rong, 2014) Illustration of two architectures of word2vec models

den layer is computed by taking average over the projected vectors. Mathematically (Rong, 2014),

$$\text{Output of the hidden layer} = h = \frac{1}{C} W^T (x_1 + x_2 + ... + x_C) \tag{3.3}$$

where W is the weight matrix between input and hidden layer. While taking average of the projected words, order of the words in the context window becomes irrelevant, so this architecture is treated as bag of words (BoW) model and the "*continuous*" term appears because of the continuous distributed representation of context (Mikolov et al., 2013b). Unlike skip-gram, here loss function is defined as :

$$E = -logp(w_O|x_1, x_2, ...x_C) \tag{3.4}$$

where w_O is target word, and $\{x_i\}_{i=1}^{C}$ are context words.

Negative Sampling

Negative sampling is a technique to increase the efficiency of the word2vec skip-gram model. While learning high quality of word representation, one-hot encoded representation of the target word is fed into the skip-gram model and output is also an one-hot encoded vector. So mathematically, if a vocabulary size is V and a word embedding size for a word is N, then the skip-gram model needs to learn two weight matrices of size $V \times N$ for the whole network (see figure 3.1a). Thus word2vec skip-gram model learns $2 \times V \times N$ parameters. Since the number of parameters depends on the vocabulary size, as the vocabulary size increases, number of parameters in

the network go up; eventually training time increases to optimize the large number of parameters. In addition, to avoid overfitting, the network requires a lot of data. So Mikolov et al. (2013a) came up with the idea of negative sampling to alleviate the issue. McCormick (2017) provides a nice explanation about negative sampling.

Negative samples are the words, which do not lie in the context of a target word in word2vec skip-gram model. So for a pair of context word and target word, instead of updating weights for all the V positions in the output vector, this extended model updates weights associated with the context word and some negative samples for those the entries in the output vector should be 0. Thus negative sampling helps in reducing the number of parameters, while updating the parameters; accelerating the efficiency of the model.

3.1.2 Global Vectors (GloVe)

GloVe is another method for learning vector representations of words. Pennington et al. (2014) noted that skip-gram like embedding method might be good on analogy tasks, however, it lacks the information from the global corpus statistics, since it operates on local context window. So, the authors of the paper (Pennington et al., 2014) came up with GloVe, by taking advantages of global matrix factorization[1] methods, such as latent semantic analysis (LSA) and local context window methods, such as the skip-gram model of word2vec (see section 3.1.1). They proposed a weighted least squares model, which is trained on global word co-occurrence matrix, X. Thus the model optimizes the below loss function,

$$J = \sum_{i,j=1}^{V} f(X_{ij})(w_i^T \widetilde{w_j} + b_i + \widetilde{b_j} - logX_{ij}) \tag{3.5}$$

where X_{ij} signifies the number of times word j appears in the context of word i. V is the size of the vocabulary of the given document. w_i and $\widetilde{w_j}$ represent the corresponding embeddings of the words i and j in the word embedding matrix W and context word embedding matrix $\widetilde{W}$ respectively. b_i and $\widetilde{b_j}$ are the biases of w_i and $\widetilde{w_j}$ respectively. The sole purpose of obtaining two sets of word embeddings W and $\widetilde{W}$ is to reduce overfitting and noise. Hence, the learned word embeddings are the sum of W and $\widetilde{W}$.

In the equation 3.5, f is the weighting function, which mitigates the drawback of giving equal importance to all the co-occurrences. All the co-occurrences do not provide valuable information. Rare co-occurring words are less informative and

[1] Global matrix factorization is a process of performing low-rank approximation on a large frequency-based matrix.

even noisy. Too frequent co-occurrences are also not informative. So below function is applied for providing a trade-off between such co-occurrences.

$$f(x) = \begin{cases} (x/x_{max})^{\alpha} & \text{if } x < x_{max} \\ 1 & \text{otherwise} \end{cases} \tag{3.6}$$

where x_{max} is some fixed constant. Pennington et al. (2014) empirically explored that $\alpha = 3/4$ provides good performance.

The authors however stated that the GloVe model is mathematically very similar to word2vec skip-gram model, despite having computational differences. Skip-gram model implicitly uses word co-occurrence statistics. Only difference is that skip-gram model optimizes cross-entropy error function, whereas GloVe optimizes least squared objective function. Pennington et al. (2014) articulated that cross-entropy error function gives too much weights to unlikely events, as a result, it does not model well the tail of the distributions, thus making the least square objective function suitable choice.

3.1.3 FastText

FastText (Bojanowski et al., 2016) is an extension to word2vec skip-gram model, which leverages the internal structure of the words for learning the vector representations; thus benefiting the word embeddings for morphologically rich languages, such as Turkish, Finnish. Since the training corpora of these type of languages do not includes all the word forms originating from same root, rare word forms are not learned very well. As a consequence, learning representations for character n-grams of the words improve the word vector representations. Each word is represented as bag of character n-grams. For example, taken from the paper by Bojanowski et al. (2016), the word "*where*" can be represented by its character n-grams for $n = 3$:

<wh, whe, her, ere, re>

This set of character n-grams also includes the word itself, but in a special form, which is "<*where*>". These special symbols, "<" and ">" are used to distinguish between the word and character n-gram of a sequence. For example, "*her*" is present in the sequence of character n-grams of word "*where*". However, word "*her*" can be present in the vocabulary. In order to distinguish between these, word "*her*" is represented as "<*her*>".

During the training process of word2vec skip-gram model, vectors for all elements of the set of character n-grams are learned. Hence, actual word is represented as

summation of all the learned representation of n-grams and specially encoded word form. Formally, it can be defined as,

$$v_w = \sum_{e \in E} v_e \tag{3.7}$$

where v_w is the embedding of the actual word, E represents all n-grams and specially encoded form of actual word itself and v_e represents the embedding of the element lying in E.

Learning the representations for character n-grams helps in generating the vector representation of the word, unseen while training. For an out of vocabulary (OOV) word, sequence of n-grams is generated first, followed by the summation of the embeddings of all the n-grams to obtain the representation for that word. Thus, fastText is beneficial for obtaining word representation for rare words efficiently. This way character level information is also effective in representing profanity and misspelled words (Athiwaratkun et al., 2018).

3.1.4 Multilingual Unsupervised and Supervised Embeddings (MUSE)

Conneau et al. (2017) proposed an algorithm, MUSE, to align monolingual embedding spaces without using any cross-lingual annotated data in an unsupervised way. Thus, semantically similar words across languages lie in the same embedding space. For example, English based word embedding space can be aligned with German based embedding space. As a result, word "*Dog* and word "*Hund*" (German translation of English word "*Dog*") will lie close to each other. This way it helps in unsupervised machine translation. MUSE learns the linear mapping from the embedding space of source language to the embedding space of target language by leveraging the adversarial training. Conneau et al. (2017) stated that a linear mapping, W is achieved by the Procrustes method (Schönemann, 1966) in such a way that a discriminator cannot differentiate between the mapped embedding space and target embedding space. Procrustes problem provides a closed form solution of the mathematical formula:

$$W^* = \arg\min_W \|WX - Y\|_F \tag{3.8}$$

$$= UV^T \tag{3.9}$$

where U and V are the matrices of left and right singular vectors of YX^T respectively. In the process of alignment, a discriminator is trained to maximize the ability to identify the source of the embedding of a word being sampled from mapped

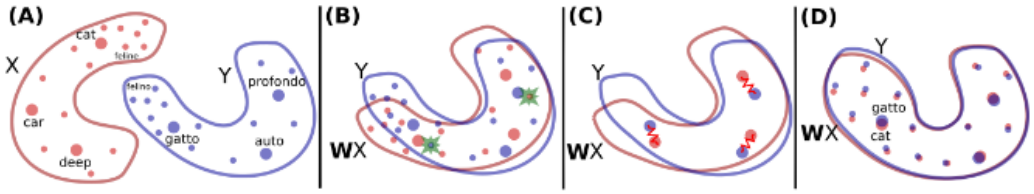


Fig. 3.2.. (Source: Conneau et al., 2017) These figures depict how MUSE works. (A) There are two embedding spaces, X and Y. The red one is for English words and the blue one is for Italian words. (B) Slight alignment of two embedding spaces takes place by applying matrix, W, which is learned by adversarial training. Discriminator tries to identify if the green star-ed word embeddings come from the same embedding space. (C) Linear mapping, W, is further modified by the Procrustes problem so that the source embedding space can be aligned effectively. (D) Finally, alignment is complete.

embedding space, whereas linear mapping, W tries to fool the discriminator, so that it cannot discriminate. This way, mapped embedding space gets close to the target embedding space. Figure 3.2 depicts the graphical visualization of MUSE.

3.2 Orthogonal Procrustes

The orthogonal Procrustes problem (Schönemann, 1966) aims to provide an orthogonal transformation matrix, W, which projects a matrix $A \in \mathbb{R}^{n \times m}$ onto a matrix $B \in \mathbb{R}^{n \times m}$, such that $\|AW - B\|_F^2$ is minimized. In other words, it transforms a point from a subspace A, closest to the corresponding point in the subspace spanned by the matrix B. The reason for choosing an orthogonal W is that Frobenius norm[2] $\|.\|_F$ is invariant under orthogonal transformation. This implies,

$$\|AW\|_F^2 = \|A\|_F^2 \tag{3.10}$$

Mathematically, the orthogonal Procrustes problem statement can be formulated as:

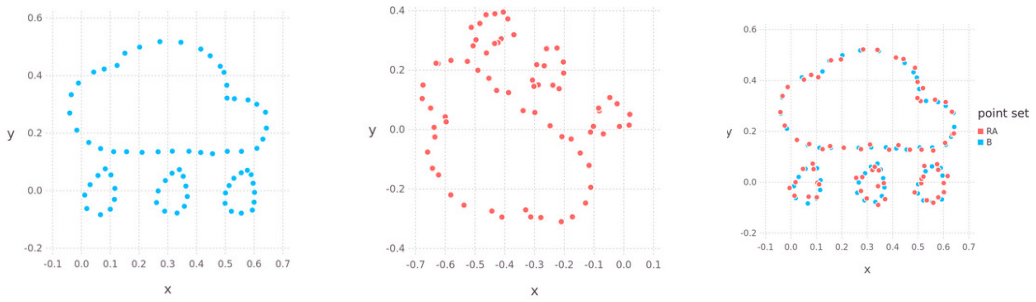
$$W^* = \arg\min_W \|AW - B\|_F^2 \quad \text{s.t.} \quad W^T W = I \tag{3.11}$$

This problem results in a closed form solution for W, which is represented as:

$$W = UV^T \quad \text{where} \quad UDV^T = \text{SVD}(A^T B) \tag{3.12}$$

U and V are the matrix of left and right singular vectors of $A^T B$ respectively. D is the matrix of singular values. Mount (2014) provides a detail derivation of the mathematical formula for better understanding. Figure 3.3 provides a visualization

[2]Frobenius norm of a matrix X is define as, $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$



(a) (b) (c)

Fig. 3.3.. (Source: Cory, 2018). These figures provide a toy example of the orthogonal Procrustes in 2-dimensional subspace. (a) Blue dots represent the points lying in a subspace, spanned by matrix B. (b) Red dots represent the points lying in a subspace, spanned by matrix A. (c) It depicts that red dots are very close to the corresponding blue dots. This implies that after orthogonal transformation of matrix A onto matrix B, points of matrix A are very close to the corresponding points of matrix B.

of orthogonal Procrustes. This process has a very good application in machine translation (Xing et al., 2015).

3.3 Canonical Correlation Analysis (CCA)

Hotelling (1936) built the concept of canonical correlation analysis (CCA) for measuring the linear relationship between two multidimensional variables. Meloun and Militky (2011) treated CCA as an extension of multiple regression and correlation analysis. Multiple regression analysis finds the best 1-dimensional subspace for projecting the given multidimensional points, such that maximum correlation is achieved between given the 1-dimensional points and the projected 1-dimensional points.

However, in the case where both the given data are multidimensional, CCA is employed for achieving maximum correlation between two vectors or matrices. In the simple version of CCA, it projects the two multiple dimensional points onto two separate 1-dimensional subspaces, so that the projected points express maximum correlation with each other. In mathematical terms (Borga, 2001), given two N-dimensional variables, x and y, CCA looks for a basis for x and y separately, which is represented by w_x and w_y respectively. Therefore the projected point can be defined as:

$$\hat{x} = x^T w_x \tag{3.13}$$

$$\hat{y} = y^T w_y \tag{3.14}$$

Hence, as a next step, correlation between the two projected points, $\rho(\hat{x}, \hat{y})$ should be maximized, where

$$\rho = \frac{E(xy)}{\sqrt{E(x^2)E(y^2)}} \tag{3.15}$$

$\hat{x}, \hat{y}$ are called *canonical variates*, E is the expectation and ρ is called the *canonical correlation*. However, in general, CCA looks for more than one basis for each variable. Theoretically, cardinality of the two sets of basis is the minimum number of dimensions of x and y. This method has been used in different NLP tasks, such as sentiment analysis and product recommendation (Faridani, 2011), drawing relations between bio-medical entities (Song et al., 2018), multi-view learning of word embedding (Dhillon et al., 2011). Borga (2001) provides a detail understanding and derivation of CCA.

3.4 Retrofitting

Unsupervised learning of word representations in large corpora is semantically informative. These representations however capture abstract semantic associations and not the precise semantic relations (Glavas and Vulic, 2018). For example, Glavas and Vulic (2018) noted that recognizing synonym from antonyms in the learned vectors space is difficult; thus necessitating the refinement of the learned word vector space. This process is called retrofitting. It leverages the external lexical knowledge from lexical resources. There are two methods for retrofitting:

- Joint optimization of learning objective of the original word embeddings with the use of external linguistic constraints (Yu and Dredze, 2014).

- Retroactively refine the learned word embeddings to meet the external linguistic constraints (Faruqui et al., 2014).

3.5 Autoencoder

Autoencoder is a neural architecture for learning a lower dimensional representation of the high dimensional data point, such that the input data point can be almost reconstructed from its lower dimensional representation (Tschannen et al., 2018). An autoencoder consists of two models, encoder and decoder. Encoder is responsible for down-projecting the input data into lower dimension, whereas decoder thrives to regenerate the original input data from that of lower dimensional representation.

Mathematically, given an input data space, $\mathcal{X}$ and a lower dimensional latent space, $\mathcal{Y}$, an encoder function, f and a decoder function g are defined as:

$$f : \mathcal{X} \to \mathcal{Y} \tag{3.16}$$

$$g : \mathcal{Y} \to \mathcal{X} \tag{3.17}$$

The neural network of the autoencoder architecture learns these mapping functions, f and g by minimizing the reconstruction loss, $\mathcal{L}(x, g(f(x)))$, where $x \in \mathcal{X}$, such that:

$$f^*, g^* = \arg\min_{f,g} \mathcal{L}(x, g(f(x))) \tag{3.18}$$

$$= \arg\min_{f,g} \|x - g(f(x))\|_2^2 \tag{3.19}$$

where $\|.\|_2$ is the L_2 norm[3]. Figure 3.4 illustrates the neural architecture of the said mathematical model of the autoencoder, where subspace, $\mathcal{X}$ represents the input layer and the output layer both and subspace, $\mathcal{Y}$ represents the hidden layer. In neural network, learning mapping functions across input layer, hidden layer and output layer is equivalent to learning weight matrices W and $W^{'}$ in place of f and g respectively. W is multiplied with the input data x while downprojecting the data onto hidden layer and decoder multiplies $W^{'}$ with the hidden variable, $f(x) = Wx$ to get output, $x' \in \mathcal{X}$ with the hope that $x' \approx x$. This is the core architecture of linear autoencoder. This implies that the equation 3.19 can be written as:

$$W^*, W^{'*} = \arg\min_{W,W'} \left\| x - W^{'} W x \right\|_2^2 \tag{3.20}$$

Since the loss function in this autoencoder deals with simple matrix matrix multiplication, it is called linear autoencoder. However, hidden layer in the autoencoder can incorporate non-linear activation function, such as sigmoid (Sharma, 2017), ReLU (Sharma, 2017), on top of hidden variable, then the model is no longer linear and can be used to find out latent representation from the non linear input space. Loss function for this model can be written as:

$$W^*, W^{'*} = \arg\min_{W,W'} \left\| x - \sigma(W^{'} \sigma(W x)) \right\|_2^2 \tag{3.21}$$

where σ represents sigmoid function.

[3] L_2 norm of a vector, x is defined as : $\|x\|_2 = \sqrt{\sum_i x_i^2}$

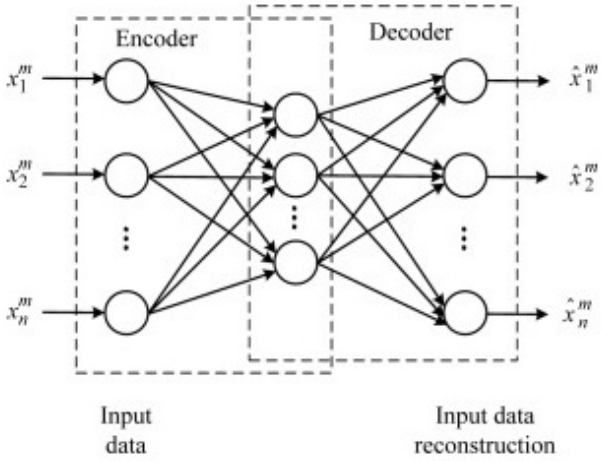


Fig. 3.4.. (Source: Lei, 2017) This figure represents the neural network of an autoencoder model, which consists of encoder and decoder model. First layer is an input layer, middle layer is the hidden layer and the last layer is the output layer, which tries to regenerate the data in the input layer.

3.6 Attention Mechanism

Bahdanau et al. (2015) came up with the attention mechanism in neural machine translation to alleviate an issue of the then popular model, basic encoder-decoder architecture. Prior to the attention mechanism, encoder in the neural machine translation encoded the input sentence into a fixed size context vector, decoder then translated the sentence from the context vector (see figure 3.5a). This resulted in compressing whole information of a sentence into fixed size vector, which is troublesome as sentence length increase. Even performance of a simple encoder-decoder based neural machine translation decreases as length of the input sentence increases (Cho et al., 2014; Bahdanau et al., 2015). Hence, to increase the performance, Bahdanau et al. (2015) extended the encoder-decoder architecture by leveraging the fact that while predicting a word in decoder, all the words of the source sentence are not relevant. Therefore, the authors thought of giving importance to the most relevant words for understanding the source sentence. This is called *attention*. As a consequence, instead of encoding whole sentence into fixed size vector, this model only encodes a subset of words of the source sentence; thus avoiding the problem of having long sentence.

In this attention-based neural machine translation model, while computing the hidden state for the current word, the decoder uses previous hidden state, predicted word in the previous node and the weighted combination of the hidden states of the encoder such that (see figure 3.5b):

$$s_i = f(s_{i-1}, y_{i-1}, c_i) \quad s.t. \quad c_i = \sum_j \alpha_{ij} h_j \tag{3.22}$$

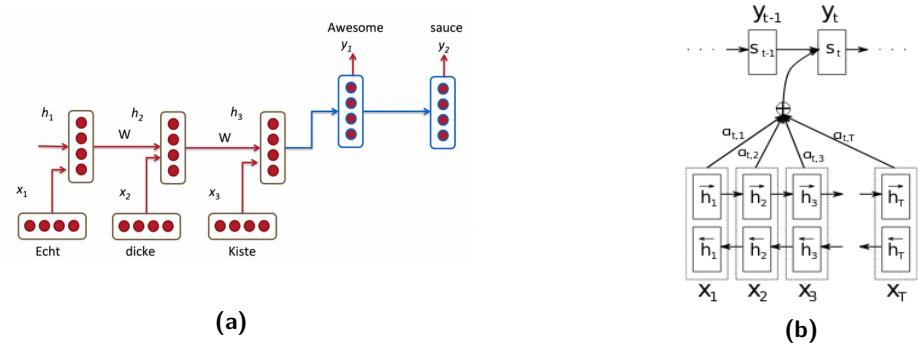


Fig. 3.5.. This illustrates neural machine translation model. (a) shows the architecture of simple encoder-decoder model for machine translation. It depicts that last hidden state of encoder encodes the whole sentence and is used as an input to decoder (source: Britz, 2016). (b) shows that how attention mechanism is used over all the hidden states of the encoder to predict the t-th word (source: Bahdanau et al., 2015).

where s_{i-1}, y_{i-1} and c_i are previous hidden state, previous predicted word and context vector respectively. α_{ij} denotes the attention for the j-th hidden state of the encoder while computing i-th context vector for predicting the corresponding word. α_{ij} can be defined as:

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_k exp(e_{ik})} \quad s.t. \quad e_{ij} = a(s_{i-1}, h_j) \tag{3.23}$$

where e_{ij} is called *alignment model*, which measures the suitability of word j for predicting the word i in decoder. Bahdanau et al. (2015) used single hidden layer feed forward network for learning alignment model in their proposed architecture. The authors also learned the weights of this network jointly with other components of the proposed model. Weng (2018) showed the mathematical formula for obtaining the alignment model.

$$a(s_{i-1}, h_j) = W_2^T tanh(W_1[s_{i-1}; h_j]) \tag{3.24}$$

where $[s_{i-1}; h_j]$ denotes the concatenation of s_{i-1} and h_j in the input layer, $tanh$ is the activation function in the hidden layer and weights W_2 and W_1 are learned in the alignment model. Luong et al. (2015) showed some other methods for measuring the alignment score, such as:

$$a(s_{i-1}, h_j) = \begin{cases} h_j^T s_{i-1} & \text{dot product} \\ h_j^T W s_{i-1} & \text{general} \end{cases} \tag{3.25}$$

where W is the learned weight in the network. Learning the alignment score also helps in visualizing how the models work (Weng, 2018) along with understanding the context of the sentence efficiently.

3.7 Evaluation Metrics

Topic modeling is a technique for generating topic or aspect from a document. A topic in a document is inferred based on the related words lying in a cluster referring to that topic. So an important task in topic modeling is that topic should express related words in the corresponding cluster. Since, there are different methods for topic modeling, such as LDA and ABAE, understanding how the methods perform on the different tasks and data is very much needed. So there should be a metric to measure how the words in a cluster cohere with the topic or aspect representing that cluster. There are several metrics ranging from automatic intrinsic measurement to manually crafted extrinsic measurement (Stevens et al., 2012). In this thesis, we have used word co-occurrence based and word vector similarity based evaluation metrics. So we have focused only on these two type of methods in this section.

3.7.1 Word Co-occurrence Based Coherence Score

Topic modeling or aspect term clustering techniques generally cluster the semantic aspect terms. So word co-occurrence based evaluation methods consider two words from a cluster each time and measure the co-occurrence of these two words in either external corpora or the corpora topic model is trained on. However, several such evaluation metrics have been proposed in order to meet the maximum correlation criteria with the human judgment on the coherence of the clusters. For example, Stevens et al. (2012) noted that the UCI evaluation method and UMass evalution method go well with human judgment (Newman et al., 2010; Mimno et al., 2011).

UCI Evaluation Metric

Newman et al. (2010) came up with a simple mathematical formula, based on word co-occurrence matrix to measure the topic coherence. Considering two words out of a topic or cluster, mathematical formula can be expressed as (Stevens et al., 2012):

$$score(w_i, w_j) = log\frac{p(w_i, w_j) + \epsilon}{p(w_i)p(w_j)} \tag{3.26}$$

where p measures the probability of the word occurring in a document of some external reference corpora, such as Wikipedia. ϵ is used to avoid the computation of $log(0)$, which might occur if words w_i and w_j do not co-occur in the reference corpora. The coherence score for a topic or an aspect V is defined as (Stevens et al., 2012):

$$\text{coherence-score(V)} = \sum_{(i,j)\in V} score(w_i, w_j) \tag{3.27}$$

where V is a set of words describing the corresponding topic. Since this evaluation metric accesses external references, this is called extrinsic evaluation method. In other words, this metric measures the semantic similarity between words with respect to some reference corpora. This way, Newman et al. (2010) measures the coherence of topics automatically.

The authors, in order to observe how this metric correlates with human judgment, took help from some annotators, who rated the topics on a 3-point scale (Lau et al., 2014).

UMass Evaluation Metric

UMass metric (Mimno et al., 2011) is another variant of evaluation metrics, which is also based on words co-occurrence. Unlike UCI metric, mathematical formula for this metric is defined as (Mimno et al., 2011):

$$score(w_i, w_j) = log\frac{D(w_i, w_j) + 1}{D(w_j)} \tag{3.28}$$

where $D(w_i, w_j)$ represents the number of documents wherein words w_i and w_j co-occur. $D(w_j)$ measures the number of documents consisting of word w_j. 1 is used as a smoothing count to avoid the computation of $log(0)$ like UCI metric, which might occur if words w_i and w_j do not co-occur. The coherence score for a topic or an aspect, V is defined as (Mimno et al., 2011):

$$\text{coherence-score(V)} = \sum_{i=2}^{M}\sum_{j=1}^{i-1} score(w_i, w_j) \tag{3.29}$$

where topic, V contains top M most relevant words. However, unlike UCI metric, this metric accesses the corpora on which topic model is trained while counting the word occurrence. Thus this metric is intrinsic and it reflects how well the topic model captures the training corpora. The authors also found that this method correlates well with the human perception. They performed the similar steps performed by Newman et al. (2010) for measuring the correlation with human judgment.

Normalized Point-wise Mutual Information (NPMI)

This metric (Bouma, 2009) is also another variant, which is based on word co-occurrence matrix. The point-wise mutual information measures how much the

probability of two words co-occurring differs from their individual occurrence in the corpora (Bouma, 2009). Mathematically it is defined as:

$$PMI(w_i, w_j) = log\frac{p(w_i, w_j)}{p(w_i)p(w_j)} \tag{3.30}$$

where $p(w_i, w_j)$ represents the probability of words w_i and w_j co-occurring in a document. $p(w_i)$ and $p(w_j)$ measures the probability of words w_i and w_j occuring individually in a document. NPMI is a normalized version of point-wise mutual information, which is used to have a fixed upper bound (Bouma, 2009). It is written as:

$$NPMI(w_i, w_j) = \frac{log\frac{p(w_i,w_j)}{p(w_i)p(w_j)}}{-logp(w_i, w_j)} \tag{3.31}$$

Hence, for a topic, V containing M topic words, NPMI can be defined as:

$$\text{coherence-score(V)} = \sum_{i=2}^{M}\sum_{j=1}^{i-1} NPMI(w_i, w_j) \tag{3.32}$$

Lau et al. (2014) showed that NPMI has a strong relationship with human perception.

3.7.2 Word Vector Similarity-based Coherence Score

In section 3.1, we have learned that word embedding techniques learn vector representations for words so that semantically similar words lie close to each other. This implies that if we measure the cosine similarity between the vector representations of two words, then relatedness between two words can be captured. Thus coherence score of a topic can be calculated by measuring cosine similarity of each pair of words from that topic. This is called word vector similarity based coherence score.

Since word embedding algorithm explicitly or implicitely uses word co-occurrence statistic and NPMI also takes advantages of word co-occurrence matrix, Ding et al. (2018) expressed nice resemblance between NPMI and word vector similarity based coherence score. The authors even showed experimentally that GloVe based vector similarity correlates well with human perception like NPMI. Mathematically, word vector similarity based coherence score can be defined as (Ding et al., 2018):

$$\text{coherence-score(V)} = \frac{1}{N(N-1)} \sum_{i=2}^{N} \sum_{j=1}^{i-1} \langle E_{i,:}, E_{j,:} \rangle \tag{3.33}$$

$$= \frac{\sum\{E^T E\} - N}{2N(N-1)} \tag{3.34}$$

where $\langle \cdot, \cdot \rangle$ represents inner dot product. V denotes the topic, which contains N topic words. $E \in \mathbb{R}^{N \times D}$ represents the embedding matrix of those N topic words, such that $\|E_{i,:}\| = 1$, where $E_{i,:}$ represents row of the matrix, E. This method is indeed computationally efficient (Ding et al., 2018).

4 Method

This chapter outlines ABAE model, which is the core architecture of our thesis. This chapter also explains why we chose ABAE model. Furthermore, it explores how retrofitting (see section 3.4 in chapter 3) is used for fine-tuning the GloVe pre-trained embeddings. Note that the detail description of ABAE model has been taken from the paper, written by he et al. (2017)

4.1 Topic Modeling

Topic modeling is an effective unsupervised way for extracting aspect terms and categorizing them into aspects. An aspect[1] is an inferred label for a cluster of aspect terms or topic words. Categorization helps in inferring an aspect. LDA-based models (Blei et al., 2002) have been predominant in this task. However, LDA uses bag of words, so topic words do not need to be coherent (Schouten et al., 2018). he et al., 2017 also noted that LDA-based model does not categorize semantically similar words into same cluster to a large extent. Therefore, to improve the clustering, he et al. (2017) proposed an attention-based model, which is ABAE. They showed that ABAE model outperformed LDA in their task. This explains why we chose ABAE model for our thesis.

4.1.1 ABAE Model

ABAE model is an attention-based autoencoder (discussed in section 3.5) model, proposed by he et al. (2017). ABAE model aims to learn an aspect by finding the nearest words to the aspect representation in word embedding space (see section 3.1); thus leading to the generation of a set of aspects.

[1] An aspect can also be considered as a group of aspect terms. Understanding the difference between aspect term and aspect is important for rest of our thesis.

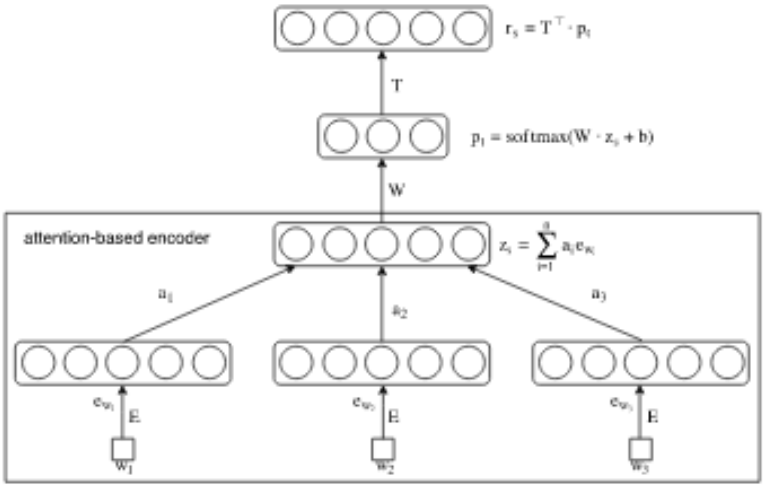


Fig. 4.1.. (Source: he et al., 2017) This figure illustrates the architecture of ABAE model.

General Overview of the Model

Given a corpus, ABAE model generates a word embedding space $E \in \mathbb{R}^{V \times D}$ by using word2vec algorithm (Mikolov et al., 2013b) (see section 3.1.1) with the hope that a word representation $e_w \in \mathbb{R}^D$ of a word w captures the semantic relation with other words of the corpus. V is the number of vocabulary in the corpus. These representations of words help in forming meaningful sentence representation. Since an aspect is inferred from a group of aspect words and these words are extracted from the same embedding space E, it can be perceived that aspects are sharing same embedding space. This implies that an aspect can also be represented in the space E. As a consequence, if a corpus exhibits K aspects, where $K << V$, then the model constructs an aspect embedding matrix $T \in \mathbb{R}^{K \times D}$. Since extracting nearest words around aspects in space E is meaningful for understanding aspects, so learning this matrix T is essential.

Given an input sentence and an indexed vocabulary, a sentence is represented by a set of indices based on its words' position in the vocabulary. This set is then fed to ABAE. In the encoder section, attention mechanism (discussed in section 3.6) plays an important role in identification of aspect terms. It assigns more weights to the words relevant to the context of the sentence and provides less weights otherwise. This mechanism is employed to get close to the aspect terms. Linear combination of these weighted embeddings of the words leads to the sentence representation in vector from z_s. In the decoder section, ABAE model tries to regenerate the input sentence representation with minimum information loss by exploring the row space of the aspect embedding matrix T. Figure 4.1 represents the autoencoder architecture of the model.

Attention Mechanism in Sentence Representation

In the section 3.6, we have discussed the role of attention mechanism in basic encoder-decoder architecture and how it is useful in representing the context of a sentence than encoding the whole sentence. This motivated the authors to use the attention layer in the model to take advantage of the most relevant words of the input sentence with respect to aspects while representing the sentence. As discussed above, input sentence is represented as a linear combination of weighted word embeddings, so considering a sentence having n words, z_s can be represented as:

$$z_s = \sum_{i}^{n} \alpha_i e_{w_i} \tag{4.1}$$

where α_i represents the weights associated with words. This weight is measured by the attention layer in the model as discussed in section 3.6. Mathematically, weights α_i can be written as:

$$\alpha_i = \frac{exp(d_i)}{\sum_k^n exp(d_k)} \tag{4.2}$$

where d_i is referred as alignment score. As per our discussion in section 3.6, there are different ways to calculate the alignment score, such as inner dot product, using a matrix M as parameter in the network (see equation 3.25). Another method is deploying feed forward network having one hidden layer. The authors applied the second approach by using a matrix M. So d_i can be define as:

$$d_i = e_{w_i}^T M y_s \tag{4.3}$$

where y_s is simple average of the embeddings of the words of the sentence, such that:

$$y_s = \frac{1}{n} \sum_{i}^{n} e_{w_i} \tag{4.4}$$

The matrix $M \in \mathbb{R}^{D \times D}$ in equation 4.3 is learned while training the model. The authors considered y_s as global context embedding of the sentence, so they perceived the matrix M as a mapping between the global context embedding of the sentence and the word embedding. This matrix helps in finding the relevance of the words towards the sentence.

Reconstructing the Sentence Embeddings

We have discussed so far how attention is given to words of a sentence and how it is effective in representing an input sentence. In this section, we outline how the decoder reconstructs the sentence embedding and the different important compo-

nents for that purpose. We stated above that reconstruction of the input sentence embedding explores the row space of aspect embedding matrix T, which implies that linear combination of aspect embeddings regenerates the sentence embedding. Mathematically, given the reproduced sentence embedding r_s, it can be defined as:

$$r_s = T^T p_t \tag{4.5}$$

where $p_t \in \mathbb{R}^K$ represents the weight vector. In words, p_t is a probability distribution over K aspects, which denotes the likelihood of an input sentence belonging to a certain aspect. For example, p_{t_i}, where $i \in K$, represents the probability that the input sentence belongs to i-th aspect. Mathematical relation between input sentence embedding z_s and the probability distribution p_t is defined as:

$$p_t = softmax(W z_s + b) \tag{4.6}$$

where W, b are learned while training the model and softmax function (as described by Bendersky, 2016) $\mathcal{S}$ is written as:

$$\mathcal{S} : \begin{pmatrix} a_1 \\ a_2 \\ . \\ . \\ a_K \end{pmatrix} \mapsto \begin{pmatrix} S_1 \\ S_2 \\ . \\ . \\ S_K \end{pmatrix} \quad s.t. \quad S_j = \frac{exp(a_j)}{\sum_{i=1}^{K} exp(a_i)} \tag{4.7}$$

where $\sum_{j=1}^{K} S_j = 1$. To summarize the the reconstruction process, decoder learns the parameters represented by T, W and b during the training phase, so that the sentence embedding r_s can be reconstructed as close as z_s.

Training ABAE Model

Every neural network is trained on an objective function for learning the parameters of the model. ABAE is also not an exception. Therefor, ABAE model optimizes a loss function to achieve $r_s \approx z_s$. Since this model is built upon autoencoder, the model requires a reconstruction error function (discussed in 3.5) to learn the optimal values for the parameters $\{T, M, W, b\}$. The authors used the similar concepts in this model as described in section 3.1.1 about negative sampling with regards to word representations. This model randomly samples m sentences for an input sentence such that the reconstructed sentence embedding r_s can no way be close to those m sentences. Thus the input sentence is reconstructed efficiently and the corresponding reconstruction loss can be represented as hinge loss (Rosset et al., 2003) which

maximizes the similarity between r_s and z_s and minimizes the similarity between r_s and representation of negative samples. The loss function is given by:

$$\mathcal{J} = \sum_{s \in D} \sum_{i=1}^{n} max(0, 1 - r_s z_s + r_s n_i) \tag{4.8}$$

where D, s and n_i denote a corpus, an input sentence and average word embeddings of the words of i-th negative sample respectively. This error function is also called contrastive max-margin objective function since it maximizes the margin between correct pair of input-output (z_s and r_s) and incorrect pair of input-output (n_i and r_s). This function was also used by Weston et al., 2011, Socher et al., 2014, Iyyer et al., 2016 in their works.

Since learning aspect embedding matrix T is the heart of the training, capturing diverse aspects out of the corpus instead of redundant aspect is very much beneficial. So the authors employed a regularization term with the hope that the model will learn the aspect embedding matrix T in such a way that dot product between any two aspect embeddings is close to zero. This ensures the learning of different aspects from the corpus. Therefore, the suitable regularization term should be:

$$\mathcal{U} = \left\| T_n T_n^T - I \right\|_F \tag{4.9}$$

where $\|.\|_F$ represents Frobenius norm. $I \in \{0, 1\}^{K \times K}$ is an identity matrix. T_n is a normalized version of T, where norm of each row of T is 1 ($\|T_{i,:}\|_2 = 1$). Therefore, the regularized loss function can be written as:

$$\mathcal{J}_{reg} = \mathcal{J} + \lambda \mathcal{U} \tag{4.10}$$

where λ is a hyperparameter to the model and it is defined as regularization weight. As a result, ABAE model optimizes loss function $\mathcal{J}_{reg}$ and learns meaningful aspects out of a given corpus.

4.2 Mittens: Fine-tune Pre-trained GloVe Embedding

This section of thesis describes how Mittens perform fine-tuning on pre-trained GloVe embedding, which is important for the subsequent sections. However, the importance and the need of this framework in this thesis will be explained in detail in experiment section.

Pre-trained GloVe generally captures the semantic relations among words in a

large voluminous corpora, such as Wikipedia. Certainly, this knowledge does not master the semantic relations in a domain specific corpora. So, Dingwall and Potts (2018) came up with Mittens framework to perform fine-tuning on pre-trained GloVe embedding to capture domain specific semantic relation. Mittens utilizes the concept of retrofitting (discussed in section 3.4) in the process. Mittens modifies the embedding space learned by GloVe on domain specific corpus so that the learned representation lies close to pre-trained embedding; thus taking advantage from external corpus' knowledge constraints. Therefore, the authors extended GloVe's objective function and proposed a new one (taken from the authors' paper):

$$\mathcal{J}_{mittens} = \mathcal{J} + \mu \sum_{i \in R} \|\hat{w}_i - r_i\|_2^2 \tag{4.11}$$

where $\mathcal{J}$ denotes the objective function (see equation 3.5) for GloVe embedding algorithm. R represents the words present in both vocabulary V and $V^{'}$, where $V^{'}$ denotes the vocabulary for the pre-trained GloVe embedding and V is the vocabulary for the domain specific corpus. $\hat{w}_i$ is the glove embedding, which is obtained while optimizing the function $\mathcal{J}$ on domain specific corpus and r_i is the pre-trained embedding. μ is a non negative real-valued weight, applied to the difference between the embeddings of the common vocabulary present in both corpora (domain specific and large generic corpora).

Since the objective function of Mittens optimizes the GloVe objective function $\mathcal{J}$, it not only brings learned vector representations of the common vocabulary close to the pre-trained GloVe embeddings, ensured by optimizing $\mu \sum_{i \in R} \|\hat{w}_i - r_i\|_2^2$, but also optimizes representations of other words of the domain specific corpus without distorting words co-occurrence statistics. Hence, Mittens objective function ensures that the final learned representations of words capture the semantic relationship of a general-purpose corpus as well as domain specific corpus.

In the process of retrofitting, the authors modified GloVe objective function (see equation 3.5) for vectoring, so that Mittens can take full advantages of hraphics processing unit (GPU) and run faster. The vectorized GloVe objective function can be written as:

$$\mathcal{J} = f(X)M^T M \quad s.t. \quad M = W^T \widetilde{W} + b1^T + 1\widetilde{b}^T - g(X) \tag{4.12}$$

where X is words co-occurrences matrix. Columns of W and $\widetilde{W}$ represent word embedding and context embedding vectors respectively. b and $\widetilde{b}$ are the corresponding biases of W and $\widetilde{W}$. The authors described function g as:

$$g(X_{ij}) = \begin{cases} K & \text{if} X_{ij} = 0 \\ log(X_{ij}) & \text{otherwise} \end{cases} \tag{4.13}$$

where K is any constant and certainly it does not have any impact during the training process. Dingwall and Potts (2018) showed in their paper that Mittens is an efficient algorithm and simple extension to GloVe embedding, whose resultant embedding has proved to be very effective in different tasks.

5 Data

This thesis explored organic non annotated dataset for conducting a set of experiments to find out aspects and distribution of data over the aspects. Therefore, in this chapter, we describe our corpus and the corpus statistics.

5.1 Organic Non Annotated Dataset

Since our domain of interest lies on organic food data, we thought of exploring the insights in human perception about organic food. So we worked on the data collected by Widmer (2018) and Jain (2018). They gathered the user generated comments, which is related to organic food, its benefits, farming and life styles from different well known editorials, blogs, news article etc. for German and English. Kasischke (2019) provided a nice summarization of their data collection process. Below figure depicts a tree diagram of different sources of organic food data, such as biased and unbiased.

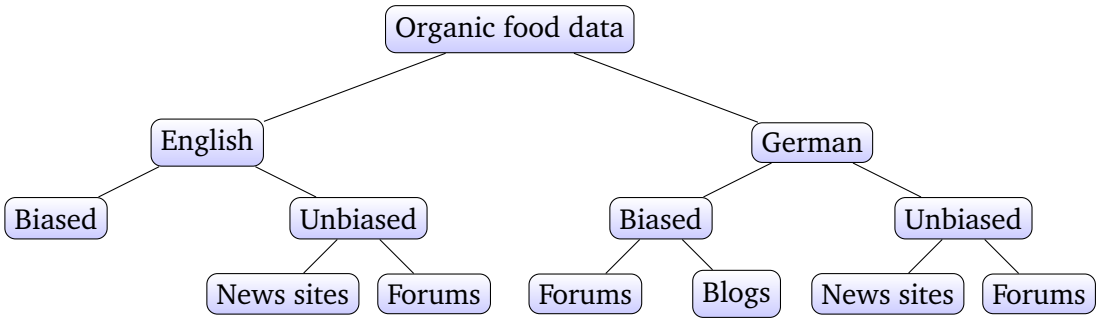


Fig. 5.1.. This diagram illustrates different source of organic food data for English and German.

Biased data are generally biased user generated opinions. For example, users' comments over some organic products' websites are positive to a large extent due to their inclination to organic products. On the other hand, news articles' contents are considered not to be biased, because those include lot of average minded peoples' opinions and perspectives. So these are unbiased data. Hence, biased and unbiased data have subcategories e.g. forums, blogs and news sites to indicate the sources of these data. However biased organic food data in English does not have any subcategory to indicate the source of the data. These data were collected from Facebook pages, few online sites such as Organic Authority, Organic Consumers etc. Similarly, English

	German		English	
	Biased	Unbiased	Biased	Unbiased
Total comments	6165	221285	306037	209310
Relevant comments	1951	105509	86728	54208

Tab. 5.1.. This table portrays the number of relevant and total biased and unbiased comments for German and English.

news articles were gathered from New York Post, Washington Post etc. and few German blogs are Campact, Eat Smarter etc. Our javaScript object notation (JSON) data (shown in appendix A.1.1) mainly consists of two contexts, article and users' comment on that article. However, our motivation is understanding aspects of user generated comments, so this thesis focused only on the comment section of the data instead of article part. Although most of the keys, such as "*article_title*", "*article_text*" of this JSON data are self-explanatory, we would like to focus on "*relevant*" flag, as it is an important factor of our thesis. This flag represents if the article is relevant to our domain of interest, such as food or organic food. So we worked only on the data, whose relevant flag were set to 1. Table 5.1 shows the corpus statistics on which this thesis performed downstream tasks. These statistics clearly show that German unbiased relevant comments are dominant over German biased relevant comments, whereas number of English biased and unbiased relevant comments are quite at par. Hence, it is expected that our model will extract more number of aspects for German unbiased data, whereas we expect that the model will extract balanced number of aspects from overall biased and unbiased data for English.

Furthermore, we have seen that ABAE model (discussed in section 4.1.1) works on sentence level instead of comment level, since it generates sentence embedding for each sentence while learning. For that reason, we applied the next course of actions, such as preprocessing and keyword-based filtering (discussed in chapter 6) on sentences. A sentence in our database is considered as a set of characters that ends with character, "\n". English dataset contains 209186 sentences out of 140936 relevant comments (see table 5.1) and similarly German dataset contains 316296 sentences out of 107460 relevant comments (see table 5.1). We have provided more statistics for broad understanding about data in appendix section.

6 Experiments and Results

So far we have learned about our dataset, different models, methods and concepts. This chapter outlines how we utilized our data and different methods to perform a set of experiments. Furthermore, in this chapter, we report and explain the results we observed post experiments.

6.1 Data Preprocessing

User generated comments are always unstructured in nature. For understanding and finding meaningful insights from the unstructured original comments, we need to provide a structure or semi-structure to the comments (Wang, 2008; Gurusamy and Kannan, 2014). So preprocessing is an important step for that task. This section exhibits how the thesis shaped the data so that it can be fit effectively and efficiently to the model. Although few preprocessing steps are applicable to both English and German organic food data, some steps are very specific to individual corpus. For that reason, we have also stated the whereabouts of the applications of the steps in the following preprocessing steps. This thesis followed an ordered sequence of the preprocessing steps the way they are presented in the following sections. However, it is not mandatory to follow the same order, as long as the ordered sequence of preprocessing steps make sense for the data. For example, lemmatization should be performed after stop word removal or any keyword based data filtering, since lemmatization changes a word's form, so it might change keywords' form in the data. Similar consideration should also be stated for stop word removal.

6.1.1 Case-folding

Since we focused on finding aspect terms and categorizing them to meaningful aspects, so machine should understand that words "*Food*" and "*food*" are same i.e., should not be sensitive to cases. Model should not treat words "*Food*" and "*food*" as different aspect terms and produce different word embedding for these words. Therefore, our data should be stored in one case. It should not mix uppercase and lowercase letters. So, we converted all the sentences of our dataset to lowercase. We could have converted the texts into uppercase as well. There is no hard rule about which case should be followed. This preprocessing step was only applied to

Tab. 6.1.. The table (a) illustrates some contracted forms and the corresponding expanded forms, which we used for replacing the contracted terms in the corpora. Table (b) similarly exhibits some misspelled words and the associated correct forms, which were used for correcting the English dataset.

Contracted Forms	Expanded Forms
aren't	are not
could've	could have
I'm	I am

(a)

Misspelled words	Correct forms
ogranic	organic
pestecides	pesticides
thier	their

(b)

English data, since case is more sensitive in German data. For example, word "*MIT*" in German can refer to Massachusetts Institute of Technology. However, word "*mit*" in German means "*with*". So if we convert case, then word "*MIT*" and "*mit*" will be perceived as same word by the model.

6.1.2 Contraction Expansion

Users nowadays are prone to writing contracted forms than the full English sentences. For example, sentence "*I am on the way*" can be written as "*I'm on the way*". Consequently, model will think "*I am*" and "*I'm*" are different; thus causing difficulties in semantically similar aspect extraction. Therefore, we expanded such contraction in our corpus. We followed a dictionary, prepared by our research group, SocialROM, which consists of contracted forms and associated expanded forms. Table 6.1a displays a short of overview of the dictionary. This is only applicable to English.

6.1.3 Spelling Correction

Our dataset includes texts from social media, such as Facebook, Quora, and it is obvious that these sources contain informal texts. Informal texts are prone to misspellings. As a result, our dataset has several misspelled words. For example, word "*Hello*" can be written as "*Hellllo*" in these user generated comments. Consequently, machine might consider words "*Hello*" and "*Hellllo*" differently. Since the same English characters can repeat back to back maximum 2 times in a word, so we wiped out redundant letters from the words to make the machines understand semantic similarity easily (Backyard, 2017). Furthermore, we followed a spelling correction dictionary, provided by SocialROM to correct our dataset. Table 6.1b is a short illustration of that dictionary. We applied the whole step for English data only, since we lacked such dictionaries or resources for German data.

Tab. 6.2.. These tables are a short overview of the dictionaries of some special characters and abbreviations of the corpus. (a) represents such special characters, which were replaced by the corresponding words in the corpus. (b) represents such abbreviations, which were replaced by the corresponding full form in the corpus.

Special Characters	Replaced Words
%	percentage
$	dollar
+	plus

(a)

Abbreviations	Full Forms
fyi	for your information
btw	by the way
imo	in my opinion

(b)

6.1.4 Special Character and Abbreviation Replacement

We observed that our dataset contains many special characters, such as $\{\%, \$, +\}$. Since our motivation is to understand aspects over a corpus, so instead of some special characters, learning the context of those special characters in terms of words is more beneficial. Therefore, we replaced those special characters with the meaningful words. Table 6.2a gives examples about special characters.

Peoples nowadays are prone to using some social media abbreviation while conversing and hence these abbreviated forms are very popular in today's world. So, instead of extracting some abbreviations, learning meaningful words from the full form of those abbreviation is more beneficial. Table 6.2b gives examples about abbreviations. We replaced those abbreviations with the corresponding full forms in our English data. Since we lacked such resources for German data, this thesis could not apply this method to German data.

6.1.5 Keyword-based Sentence Filtering

Our dataset contains information about food, organic food as well as agriculture, farming, pesticides and lifestyles. Since we are interested in learning topics over foods or organic foods, rest of the information unnecessarily increases the volume of the data. So we filtered the dataset by omitting those information. For this reason, we chose the sentences having words or phrases "*organic*", "*food*", "*conventional*", "*gm*"[1], "*genetically modified*" in the English dataset. This new English dataset contains 69342 sentences. This thesis also performed an experiment (see in section 6.2.1) to analyze the importance of this keyword-based filtering. It was noticed that our model extracted more meaningful and coherent aspect terms in this new filtered dataset; thus inferring aspects easily. So, it can be mentioned that this new filtered dataset is enriched with all the important meaningful information. Similarly, we

[1]Word "*gm*" is a short form of the phrase "*genetically modified*".

chose the sentences having words {"*Bio*", "*bio*", "*Öko*", "*öko*"} in German dataset for our purpose. We used the word and its associated capitalized form, such as "*Bio*", "*bio*", intentionally, since case-folding was not performed in German dataset. This new filtered dataset contains 40620 sentences.

6.1.6 Non Informative Characters Removal

Users generated comments often consist of URLs. Our dataset is also not an exceptional in this context. Since URLs do not carry much information for understanding topics of the corpus, so presence of these information in the corpus does not benefit at all. Similarly, punctuation in a sentence is not important at all for finding the insights in the sentence. As a result we ripped of the URLs and punctuation from the text. The thesis performed this step on both English and German texts.

Similarly, our both English and German corpora consist of numeric digits, which do not make any sense in learning aspect terms. This explains why we removed digits from our textual data.

Furthermore, our English dataset contains few words of non Latin characters, because some users might provide review in their regional languages. Textual representation of those non Latin characters starts with $\{\backslash x, \backslash u\}$. Since we are interested in learning English aspect words from the English corpus, so we dropped those words from the English texts.

In general, ignoring these non informative data helps in reducing the corpora size, which reduces the complexity of the model. This step also prevents the generation of vector representations of non informative words; thus accelerating the processing.

6.1.7 Sentence Tokenization

A token is a meaningful element such as word, phrase, symbol of a sentence. Tokenization of a sentence splits the sentence into a set of tokens (Gurusamy and Kannan, 2014). Since stop word removal and lemmatization (both discussed below in detail) perform on word level of a sentence, so it was necessary for us to represent a sentence as a set of words. Below is an example, where tokenization of a sentence into words is illustrated.

$$\text{"I am on the way"} \Longrightarrow \{\text{"I"}, \text{"am"}, \text{"on"}, \text{"the"}, \text{"way"}\}$$

Therefore, in this step, we tokenized a sentence into a set of words and this step is applicable to both the corpus.

6.1.8 Stop Words Removal

Stop words are the words, which occur very frequently in the texts, but do not carry meaningful information for the understanding the texts (Gurusamy and Kannan, 2014). For example, English words "*the*", "*a*" and "*an*" are the very frequently occurring words. However, they do not carry much information. So these are the English stop words. Similarly, in German language, "*Die*", "*Ein*" are the high frequent words, which do not carry much information. These are the German stop words. Since it is not important for the model to understand these words, so we removed these word from both the corpora. Thus we reduced the corpora size. We used NLTK[2] provided lists of stop words for English and German languages.

6.1.9 Lemmatization

The words with inflectional endings are very common in English and German languages. Inflectional endings for English words are precisely {"*-ing*", "*-ed*", "*-s*", "*-es*"} and the corresponding forms of a word "*collect*" are {"*collecting*", "*collected*", "*collects*"}. Similarly, inflectional endings in German language are {"*-er*", "*-e*", "*-es*"} and the corresponding forms of a German word "*dies*" are {"*dieser*", "*diese*", "*dieses*"} (all of them mean word "*this*" in English). The same word changes to different forms in the texts for grammatical reasons (Manning et al., 2008). Consequently, word embedding algorithms and machine will treat each form as a different word, which leads to generation of different embedding for each form, though the root or base of the forms is same.

Lemmatization is a process of removing the inflectional endings from those forms and returns the base word or the dictionary form of the word, which is called lemma (Manning et al., 2008). Thus lemmatization helps in standardization of the texts, so that model can perform easily. It also helps in lowering the memory requirements of the model, since model does not require to memorize all the forms. As he et al. (2017) used NLTK provided wordnet lemmatizer[3] in their paper, this thesis also used the same lemmatizer for English dataset. However, for German language, this thesis explored the German model of spaCy[4] for lemmatization, since to the best of our knowledge, there is no integrated lemmatizer for German language in NLTK.

[2]NLTK is python based natural language toolkit. Standard way of removal stop words is presented in `https://www.geeksforgeeks.org/removing-stop-words-nltk-python/`

[3]Wordnet lemmatizer uses wordnet for lemmatization, which is large and free lexical database of semantic relationships between words (Prabhakaran, 2018).

[4]SpaCy models can be explored in detail in the webpage: `https://spacy.io/usage/models`

6.2 Aspect Extraction on English Data

Since we were interested to learn bilingual aspects, this thesis explored MUSE (discussed in section 3.1.4) as embedding algorithm. MUSE requires embeddings from English and German separately for embedding space alignment. So we performed a set of experiments utilizing ABAE model on English dataset for choosing an embedding algorithm.

6.2.1 Word Embedding Trained on Organic Data

This thesis could have employed word2vec algorithm (see section 3.1.1) for learning words' vector representations as used by he et al. (2017) and fed to ABAE model for clustering aspect terms. However, performance of word embedding depends on dataset. Since the thesis explored organic food data, whereas he et al. (2017) used restaurant and beer review corpora, we thought to explore other embedding methods along with word2vec to figure out which embedding extracts more coherent aspect terms on organic food data.

GloVe

In this experiment we explored GloVe (discussed on 3.1.2) embedding with ABAE model on organic food data, since it uses global corpus statistics explicitly. At the very first step, we trained GloVe[5] embedding model on preprocessed organic food data[6] for obtaining the vector representations of words in the corpus. Table 6.3a provides some significant hyperparameters' values, which we employed in this GloVe model during training. We did not generate word embedding using GloVe for the words occurring less than five times in the corpus, since VOCAB_MIN_COUNT is 5. In this thesis opted for 15 iterations for the training of GloVe model, as mentioned by MAX_ITER hyperparameter. Since WINDOW_SIZE is 15, GloVe model chose 15 context words for a target word for word co-occurrence statistic calculation. X_MAX factor for weighting function in glove model (see equation 3.6) is chosen to br 10. These values were taken from Standford NLP Github repository for GloVe, which were left unchanged. Since Pennington et al. (2014) used 300 as embedding size while comparing GloVe with word2vec, we chose that value as EMBEDDING_SIZE. We used a machine having 12 cores for our thesis, so NUM_THREADS was 12 for parallelization in GloVe model.

[5] Code of GloVe model was taken from Standford NLP Github repository, `https://github.com/stanfordnlp/GloVe`

[6] Preprocessed organic food data contains all the relevant sentences. Step 6.1.5 was not applied for the sake of a specific experiment.

Tab. 6.3.. These tables illustrate some hyperparameters of GloVe model and ABAE model.

Hyperparameters	Values
VOCAB_MIN_COUNT	5
EMBEDDING_SIZE	300
MAX_ITER	15
WINDOW_SIZE	15
NUM_THREADS	12
X_MAX	10

(a) GloVe model

Hyperparameters	Values
VOCAB_SIZE	15000
LEARNING_RATE	0.001
EPOCHS	15
BATCH_SIZE	64
NEGATIVE_SAMPLES	10
REGULARIZATION_WEIGHT	0.1
MAX_LENGTH	256

(b) ABAE model

Once we generated 300 dimensional vector representation of the words presents in corpus, we applied the ABAE model with the required hyperparameter configurations (see table 6.3b). Most of the values of hyperparameters were taken from the github repository[7] of he et al. (2017), except vocabulary size, batch size, negative samples and embedding size, which we chose to be 300. Although authors of ABAE model used 9000 as vocabulary size, we observed that vocabulary of 15000 words would lower the OOV words in preprocessed whole relevant dataset, so in this ABAE model we chose 15000 most frequent words of the corpus as vocabulary, 15 epochs and batch size of 64 for the task. The reason of different batch size than what he et al. (2017) used in their work was that batch size of 64 gave better clustering of aspect term on the corpus. Similar explanation is also applicable for choosing NEGATIVE_SAMPLES to be 10. MAX_LENGTH is responsible for clipping the sentence length if it exceeded the value. After executing ABAE model, we extracted the aspect terms of each aspect by finding the top 50 nearest words of the learned aspect vector in word embedding space using cosine similarity. This was followed by calculating the coherence scores for each aspect using the equations 3.28 and 3.29 for understanding to what extent the words are coherent in representing that aspect. Post that, average coherence score of the aspects generated by the model was calculated by averaging all the coherent score of all aspects. Thus we found out how the ABAE model performed in inferring meaningful aspects. We repeated this experiment for the aspect size ranging from 7 to 50 in order to understand the optimal number of aspects the corpus consists of.

We performed the same experimental steps as mentioned above using the same hyperparameters on the preprocessed filtered organic food data[8].

These two experiments were performed to take a decision about which dataset

[7]Github repository of the authors of ABAE model is: `https://github.com/ruidan/Unsupervised-Aspect-Extraction`

[8]Preprocessed filtered organic food data contains only relevant sentences, which matched the keywords (see step 6.1.5).

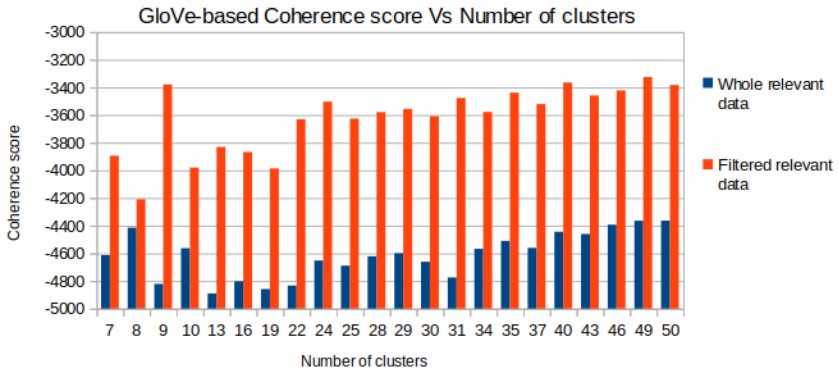


Fig. 6.1.. This figure illustrates a comparison of the aspects'/clusters' coherence scores between whole relevant dataset and filtered relevant dataset. The clusters are generated from ABAE model using GloVe embedding. The blue and red bars denote the average coherence score of the model for each aspect size ranging between 7 and 50 on the whole relevant dataset and filtered relevant dataset respectively.

we should continue with. Once the decision was taken, we optimized few hyperparameters on the chosen dataset for rest of the thesis with the hope of obtaining better aspects. Furthermore, we performed the same experiment on the chosen dataset with the optimized hyperparameter. However, while calculating the average score in this experiment, we did not use UMass coherence score (discussed in section 3.7.1), as we used earlier.

There is no unique evaluation metric for measuring coherence of aspect terms. UMass coherence score evaluation metric, which we used above is better at measuring to what extent representative words of an aspect have captured the corpus. However, we were more interested in knowing to what extent aspect terms are semantically similar to each other. Therefore, we calculated the average coherence score of each cluster/aspect by measuring word vector similarity (discussed in section 3.7.2) using the equation 3.33. In this regard, we used 300 dimensional pre-trained GloVe embeddings for aspect terms to calculate how much words are semantically similar in some external corpus. Pre-trained vectors being trained on Wikipedia and Gigaword, our external corpora for measuring coherence score were Wikipedia and Gigaword. Thus we avoided calculating that how much aspects captured the organic dataset, which is measured by UMass coherence score. We accessed *chakin*[9] downloader for obtaining the pre-trained GloVe embeddings. We discuss all the results below.

[9]chakin library is easy to use downloader for obtaining pre-trained word representations. This resource is available in `https://github.com/chakki-works/chakin`.

Results and Discussions

Figure 6.1 expresses a comparative study between aforementioned two experiments on whole relevant dataset and filtered relevant dataset. We observed in the figure that average coherence score of the aspects generated by ABAE model on filtered relevant dataset always excel over the average coherence score on whole relevant dataset. This result was found to be consistent for a range of aspect size between 7 and 50. So, it can be certainly stated that filtered relevant dataset gives more coherent aspect terms than whole relevant dataset. Therefore, we can conclude that filtered relevant corpus is more structured syntactically and better able to grasp semantic relationships. Hence, in this thesis for any further experiment we opted for the filtered relevant dataset as the corpus. Any reference of the corpus shall indicate the filtered relevant dataset hereafter.

Once the corpus was chosen, as we discussed above, we tuned few hyperparameters of ABAE model e.g. MAX_LENGTH and VOCAB_SIZE on this corpus. Figure 6.2 shows that sentences having number of words around 150 is more dominant in the corpus. This articulates that even if we clip the sentences having more than 150 words, we will not loose much information from the corpus. So we chose MAX_LENGTH to be 150 for rest of the thesis. We further noticed that vocabulary

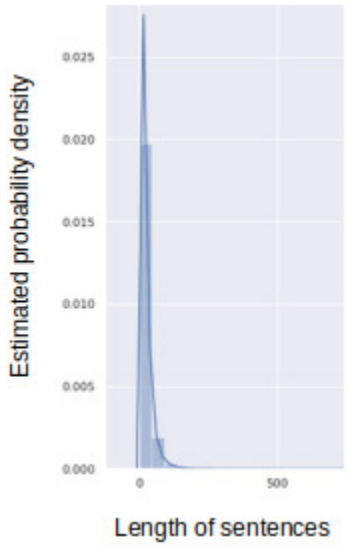


Fig. 6.2.. This figure illustrates the probability density of the length of sentences of the corpus. As the density grows, number of sentences with corresponding length increases.

size of 19000 lowered the percentage of OOV words significantly on the chosen corpus. Therefore, table 6.4 represents a set of hyperparameters for ABAE model, which we used for the remaining experiments to achieve the goal.

Hyperparameters	Values
VOCAB_SIZE	19000
LEARNING_RATE	0.001
EPOCHS	15
BATCH_SIZE	64
NEGATIVE_SAMPLES	10
REGULARIZATION_WEIGHT	0.1
MAX_LENGTH	150

Tab. 6.4.. This table outlines a set of hyperparameters for ABAE model for the rest of the experiments on the corpus.

Figure 6.3 outlines the average coherence scores of each aspect sizes ranging from 7 to 50, which were generated by ABAE model with 300 dimensional GloVe embedding trained on English organic data. We noticed that aspect size 49 gave maximum coherent clusters of aspect terms. This experiment concluded that there were 49 aspects optimally present in the corpus.

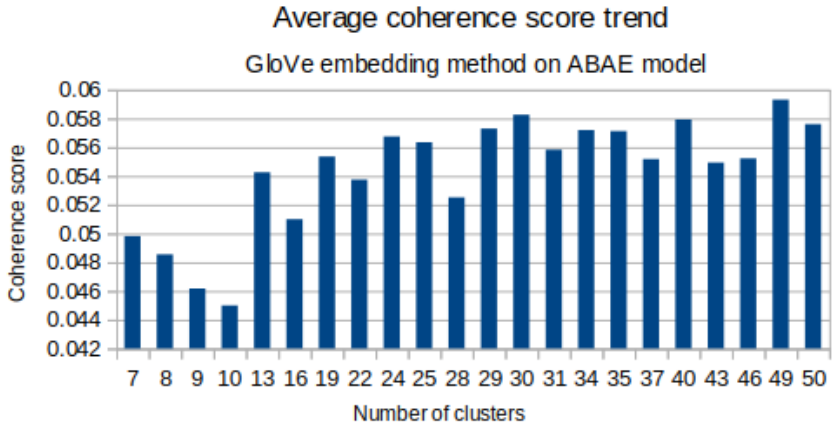


Fig. 6.3.. This figure illustrates the average coherence scores for all cluster sizes ranging from 7 to 50. These clusters of aspect terms were generated by ABAE model with 300 dimensional GloVe embeddings trained on the chosen English filtered organic data.

Word2vec

In this experiment ABAE model was applied on organic food data with word2vec (discussed in section 3.1.1) embedding model, since it is a standard and popular embedding model for learning word vector representations. We employed gensim provided word2vec[10] CBOW model, as he et al. (2017) used in their work. Since

[10]Gensim provided word2vec's implementation is available in `https://radimrehurek.com/gensim/models/word2vec.html`

Hyperparameters	Values
EMBEDDING_SIZE	300
WINDOW_SIZE	10
NEGATIVE_SAMPLE	5
VOCAB_MIN_COUNT	5
MAX_ITER	15

Tab. 6.5.. This table outlines hyperparameters' values, which were used for training word2vec model in the corpus.

word2vec model requires hyperparameters' values, we utilized most of the values (see table 6.5) provided by he et al. (2017) for learning good vector representation of the words in the corpus. However, we chose 15 iterations (see value of MAX_ITER in table 6.5), which was different from what he et al. (2017) used for learning vector representation of words. Our motivation for the experiments was finding the best suited embedding on English corpus, which was eventually a comparison among embeddings. Furthermore, Pennington et al. (2014) chose 15 iterations while comparing between GloVe and word2vec in their work. This explains why we opted for 15 iterations for training word2vec model. This explanation is also valid for embedding size 300 in this experiment. In addition, we also performed similar experiment, which we describe here to justify our choice of embedding size compared to 200, which the authors of ABAE model used.

Once word2vec model was trained completely, we fed 300 dimensional vectors for words of a sentence to the input layer of ABAE model. Thenceforth, ABAE model was run for each sentence of the corpus for generating the clusters of aspect terms out of the corpus the way we discussed above and in section 4.1.1. However, ABAE model requires some hyperparameters for learning optimal aspects from the corpus. As mentioned above, we used the values described in table 6.4 for ABAE model, so that it is quite easy to compare among experiments under the same experimental setup. After the generation of clusters of aspect terms by ABAE model, we calculated the average coherence score of each cluster/aspect by using the concept of word vector similarity (see in section 3.7.2), as we discussed in earlier experiment. For this sole purpose, we utilized 300 dimensional pre-trained word2vec embeddings, which were trained on Google News. This thesis downloaded the pre-trained word2vec embedding using chakin library. Furthermore, we conducted this experiment for aspect sizes ranging from 7 to 50 to find the optimal number of aspects present in the corpus.

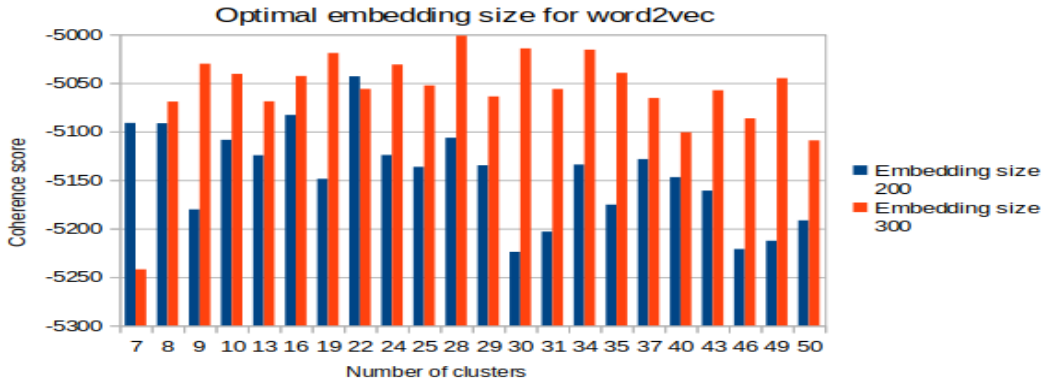


Fig. 6.4.. This figure outlines a comparison of the aspects'/clusters' coherence scores between two experiments. The blue and red bars denote the average coherence score of the model for each aspect size ranging between 7 and 50 with embedding size 200 and 300 respectively.

Results and Discussions

Figure 6.4 shows that average coherence score (calculated using the equations 3.28 and 3.29) for each aspect generated by ABAE model with vector representations of 300 dimensions always excels over ABAE model with vector representations of 200 dimensions. This observation is valid irrespective of aspect sizes ranging from 7 to 50. However, we noticed exceptions at aspect size 7 and 22 where ABAE model with 200 dimensional word2vec embedding extracted more coherent aspects than that of 300 dimensional vector representations. In addition to the need of embedding size to be 300 for the sake of comparison among embedding models, as discussed above, excellency of 300 dimensional vector representation in most of the aspect sizes motivated us to choose 300 as embedding size for this experiment.

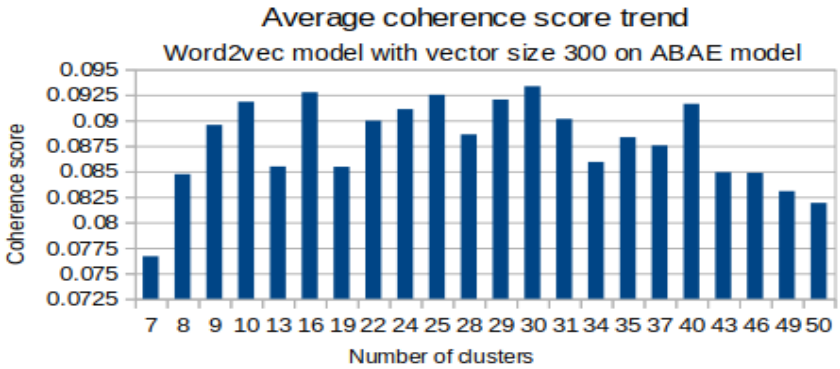


Fig. 6.5.. This figure depicts the average coherence scores of all aspect sizes ranging from 7 to 50, which were generated by ABAE model with word2vec embedding of 300 dimensions.

Figure 6.5 reflects ABAE model to what extent extracted meaningful aspect terms on the corpus and clustered them to corresponding aspects. We observed that the

average coherence score (calculated using the equation 3.33) of cluster size 30 achieved the maximum value out of the cluster sizes starting from 7 to 50. This explains that there were 30 optimal aspects within the corpus. Thus we optimized the number of aspects in the corpus with the help of coherence score.

FastText

After experimenting ABAE model with GloVe and word2vec, we formed an experimental setup for ABAE with fastText (see section 3.1.3) embedding. Since fastText takes advantages of character n-gram embedding, which leads to generation of vector for OOV words, we sensed that applying fastText on the corpus would represent OOV words more effectively and efficiently. For that reason, we began the training of gensim provided fastText[11] model on the corpus. Like GloVe and word2vec, fastText also requires hyperparameters for being trained. In this experiment, fastText utilized same values of the hyperparameters, such as WINDOW_SIZE, NEGATIVE_SAMPLE, VOCAB_MIN_COUNT, MAX_ITER, which were used (see table 6.3a) in ABAE model with GloVe trained on organic food data (see experiment 6.2.1). We also chose embedding size to be 300 due to below two reasons:

- Bojanowski et al. (2016) used 300 dimensional vector representation in their work.

- GloVe and word2vec both learned 300 dimensional vectors for words in the corresponding experiments, so we opted for 300 dimensional fastText word embedding for the sake of comparison among ABAE with different embeddings.

Since Bojanowski et al. (2016) showed that fastText skip-gram model performed better for their tasks, so we did not hesitate to turn on skip-gram mode in gensim provided implementation of fastText.

Post learning 300 dimensional vector representations, ABAE model was executed several times with those representations for the cluster sizes ranging from 7 to 50. ABAE model utilized the same hyperparameters' values, as represented in the table 6.4 for performing this experiment. After generating clusters of aspect terms, we calculated average coherence score for each cluster size starting from 7 to 50 by utilizing the idea of word vector similarity (see in section 3.7.2), as we did in earlier experiments. We employed 300 dimensional pre-trained fastText embeddings

[11] Gensim provided fastText's implementation is available in `https://radimrehurek.com/gensim/models/fasttext.html`

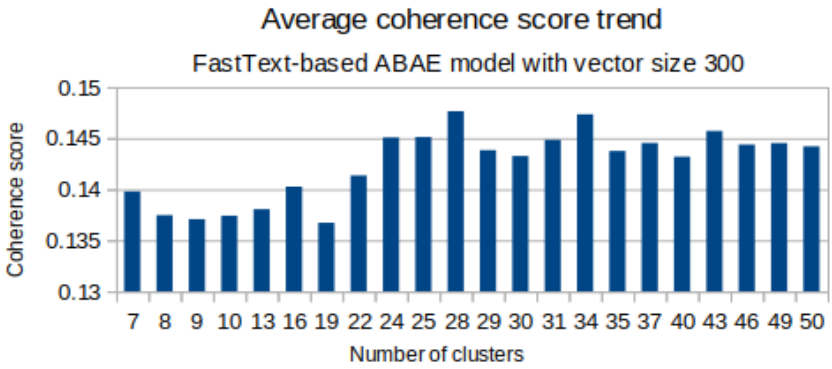


Fig. 6.6.. This figure depicts the average coherence scores of all aspect sizes ranging from 7 to 50, which were generated by ABAE model with fastText embedding of 300 dimensions trained on English organic data.

trained on English Wikipedia for the purpose of calculating coherence score. Chakin library was used for downloading the pre-trained vectors.

Results and Discussions

Figure 6.6 narrates the average coherence score of this experiment over the aspect sizes ranging from 7 to 50. Average coherence scores for aspect sizes from 7 to 22 were quite low compared to that of aspect sizes from 24 to 50. This reflects that ABAE model with fastText embedding generated more of less good quality of aspect terms for the aspect sizes ranging from 24 to 50. However, aspect size 28 produced maximum coherence score; thus generating optimal coherent aspects.

Overall Results and Discussions

We have discussed so far the results of experimenting ABAE model on English organic dataset with GloVe, word2vec and fastText embeddings trained on the corpus. Nevertheless, our motivation was finding out which embedding would perform better with ABAE model on our dataset. This led to accumulate the observations of experiments discussed in section 6.2.1. Figure 6.7 illustrates the average coherence scores for cluster sizes ranging from 7 to 50 generated by ABAE model with different embeddings, such as GloVe, word2vec and fastText. We noticed that the average coherence score for each aspect size generated by ABAE model with fastText embedding excelled over that of ABAE model with other embeddings. This observation was applicable for all the aspect sizes starting from 7 to 50. For that reason, we could conclude that fastText-based ABAE model extracted higher quality of aspects from the English organic dataset. As mentioned earlier and even noticeable in the figure

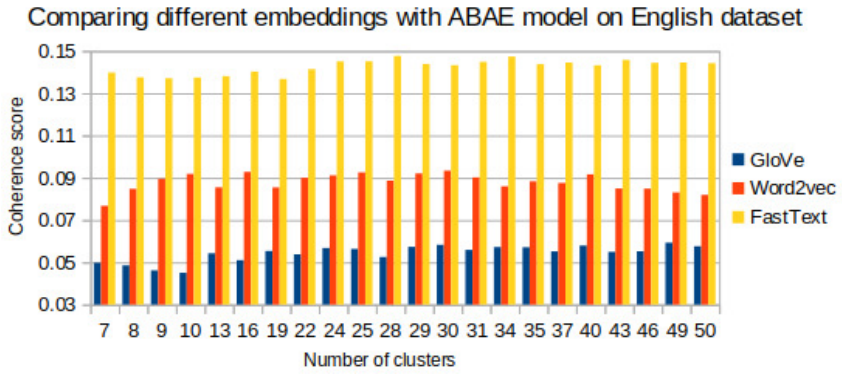


Fig. 6.7.. This figure depicts a comparison among experiments of three different embeddings-based ABAE model. These three embeddings were trained on English organic data. The yellow bar, red bar and blue bar denote the average coherence scores of all aspect sizes ranging from 7 to 50, which were generated by ABAE model with fastText embedding, word2vec embedding and GloVe embedding respectively.

6.7 that fastText-based ABAE model gave better quality of aspects at aspect size 28 within a range of aspect sizes starting from 7 to 50. Corresponding aspect terms and aspect are illustrated in appendix.

6.2.2 Fine-tuned Word Embedding and Embedding Space Alignment

Our English organic dataset contains 69342 sentences. Since embedding models require lot of parameters to be optimized, a dataset of 69342 sentences is quite small for a embedding model to learn the semantic relations. This explains the need for pre-trained word embedding model for clustering aspect terms using ABAE model, because pre-trained embedding models are trained on large voluminous dataset, such as Google News, Wikipedia etc. However, these dataset are generic corpora and our corpus is specific to organic food. Therefore, pre-trained word vectors are not capable enough in representing the contexts present in the organic food dataset. This motivated us to either fine-tuning the pre-trained word vectors or adapt the pre-trained word embedding space to embedding space trained on our corpora.

We could have performed this process on any specific word embedding model. However, as we stated in earlier experiments that our motivation was to find out which embedding model along with ABAE model perform better in extracting aspect terms. So, we describe below a set of experiments on ABAE model with different embedding models following the above mentioned ideas.

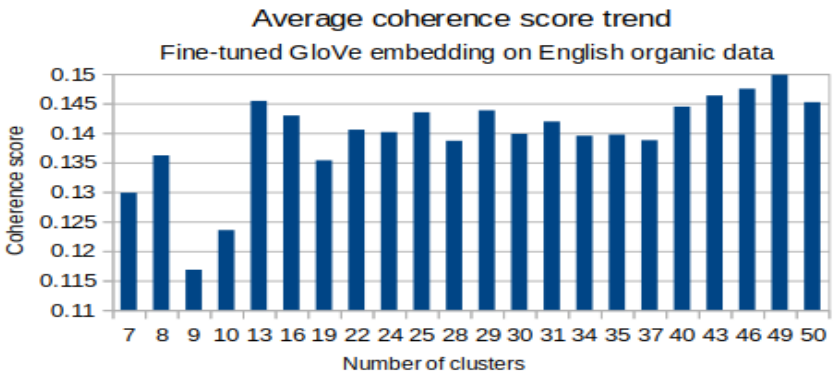


Fig. 6.8.. This figure outlines the average coherence scores for each aspect size ranging from 7 to 50. These aspects were generated by ABAE model with pre-trained GloVe embedding fine-tuned on our English corpus.

GloVe

This experiment again explored GloVe-based ABAE model, as we executed in experiment 6.2.1. However, in this experiment, we utilized 300 dimensional pre-trained GloVe embedding and then fine-tuned the vectors on our corpus, as mentioned above. At the end, we were focused to choose best suited embedding with a combination of ABAE model on the corpus, so we also chose 300 dimension for the pre-trained vectors. This helped us to compare the results among all experiments on the English organic data. We utilized chakin downloader for obtaining pre-trained word vectors trained on external corpus, such as Wikipedia and Gigaword, as mentioned earlier for the purpose of measuring word vector similarity. This is followed by fine-tuning the vectors on our corpus utilizing the framework, Mittens[12], described in 4.2.

Mittens framework requires number of dimensions of the word vectors, which was 300 in our case and maximum number of iterations required to do fine-tuning. This was 1000, as provided in the github repository. Furthermore, it needs vocabulary of our corpus and the word co-occurrence matrix, besides pre-trained GloVe vectors for the task of fine-tuning. Since co-occurrence matrix is held in main memory, so we built vocabulary of 18000 top most words out of the corpus, so that generating co-occurrence matrix would not produce out of memory error. Once the fine-tuning of 300 dimensional word vectors was complete, we applied the ABAE model with those embeddings on the corpus.

This experiment invoked the ABAE model with the hyperparameters' values mentioned in table 6.4. As a result, ABAE model generated clusters of aspect terms for an aspect size. Post that, we calculated average coherence score of that aspect using the word vector similarity, as discussed in earlier experiments. For that purpose, we

[12]How to use Mittens is available on `https://github.com/roamanalytics/mittens`

utilized the same pre-trained GloVe embedding discussed above in this experiment. We conducted this experiment for aspect sizes ranging from 7 to 50 to find the optimal aspects in the corpus.

Results and Discussions

Figure 6.8 displays the trend of average coherence scores for aspect sizes between 7 and 50, generated by above experiments. It is noticeable in the figure that average coherence scores for aspect sizes 9 and 10 are very low compared to other aspect sizes in the range. This experiment achieved maximum average coherence score at aspect size 49, which was same as we observed in experiment of ABAE model with GloVe trained in out dataset (see experiment 6.2.1). However, comparing this figure utilizing fine-tuned word embeddings with figure 6.3, we can say that this experiment gave better coherence score, not only at aspect size 49 but also at all the aspect sizes in the range. Therefore, we can conclude that fine-tuning of pre-trained GloVe embedding with ABAE model performed better than the ABAE model with GloVe trained on the our English corpus.

Word2vec

This experiment is very similar to the experiment of ABAE model with word2vec trained on our corpus (see experiment 6.2.1). However, in this experiment, we adapted the pre-trained word2vec embedding space to the embedding space learned by training word2vec model on our corpus. Adaptation was performed using alignment of embedding spaces, so that two embedding spaces, such as pre-trained embedding space and embedding space trained on our corpus would come close to each other. Thus, we were able to incorporate pre-trained vectors' knowledge constraint into context of our corpus, which was learned by our word2vec vectors.

In this experiment, alignment was achieved by two mathematical methods, such as orthogonal Procrustes (discussed in section 3.2) and CCA (see in section 3.3) to observe which one would give better results in terms of higher quality of aspects. We obtained 300 dimensional pre-trained word2vec vectors trained on Google News by using chakin library at the very first step, as we did earlier. The reason of choosing dimension to be 300 was discussed earlier. For both of the methods, we first identified the words present in both embedding space, pre-trained one and embeddings trained on the corpus. After that, we formed two embedding matrix for those common words, one was to store pre-trained word vectors and another one was to save vectors trained on the corpus. This was followed by employing

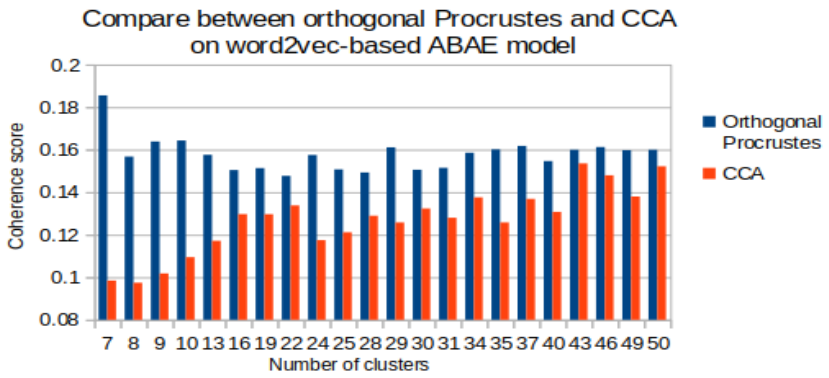


Fig. 6.9.. This figure shows average coherence scores for aspect sizes in the range between 7 and 50. The blue bars and red bars denote the scores for aspects, which were generated by word2vec-based ABAE model with the application of orthogonal Procrustes and CCA respectively.

orthogonal Procrustes[13] on these two matrices, which resulted in 300 dimensional word vectors aligned with the embedding space learned on the corpus. Post that, this experiment applied ABAE model with the aligned vectors on the corpus. Once ABAE generated clusters of aspect terms for all cluster sizes ranging from 7 to 50, we calculated the average coherence score for each cluster size by utilizing the concept of word vector similarity, as we did earlier. For that purpose, we used the same 300 dimensional pre-trained word2vec vectors as mentioned above.

Similar experiment was carried out, where instead of orthogonal Procrustes we utilized CCA for aligning both the embedding spaces (discussed above). CCA[14] only requires the number of dimension, which was 300 in our case and maximum number of iterations for the alignment process, which we chose to be 1000. Post alignment process, CCA provided two new embedding matrices, one for pre-trained vectors and another one for vectors obtained on our corpus. At the end, we took the average of this two new embedding matrices to obtain the resultant embeddings, which we utilized to invoke ABAE model with. This was followed by generation of clusters of aspect terms and calculation of average coherence score.

Results and Discussions

Figure 6.9 displays that average coherence scores for aspects, generated by the experiment involving orthogonal Procrustes method for aligning pre-trained word2vec vectors is more than that of experiment involving CCA method for alignment. This is also valid for all the aspect in the range between 7 and 50. So, we can infer that

[13]Implementation of Procrustes was taken from sklearn package and available on `https://docs.scipy.org/doc/scipy/reference/generated/scipy.spatial.procrustes.html`

[14]Implementation of CCA was taken from sklearn package and available on `https://scikit-learn.org/stable/modules/generated/sklearn.cross_decomposition.CCA.html`

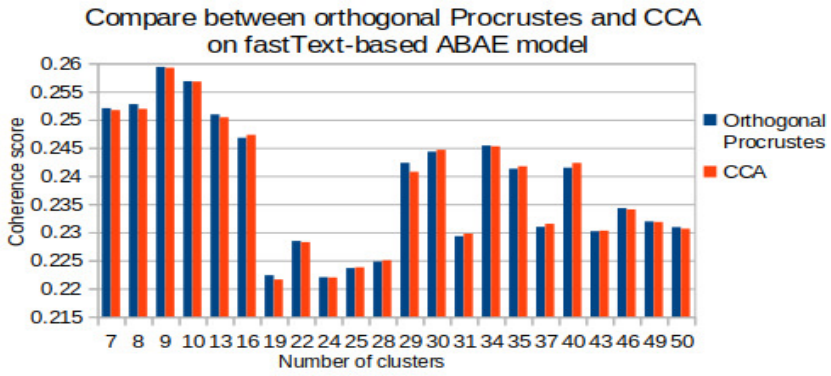


Fig. 6.10.. This figure displays average coherence scores for aspect sizes ranging from 7 to 50. The blue bars and red bars denote the scores for aspects, which were generated by fastText-based ABAE model with the application of orthogonal Procrustes and CCA respectively.

ABAE model generated better quality of aspects, when orthogonal Procrustes was applied on pre-trained word2vec vector space for the alignment than that of CCA applied on pre-trained word2vec vector space.

Comparing this figure with figure 6.5, we can even state the ABAE with alignment process of pre-trained vectors gave better average coherence score than the ABAE model with word2vec vectors trained on the corpus. For that reason, we can ensure that alignment of pre-trained word2vec vectors works better than training word2vec on the corpus in the context of applying ABAE. This figure illustrates that aspect size 7 gave optimal aspects in the corpus, when we applied ABAE with pre-trained word2vec vectors, followed by alignment process.

FastText

This experiment followed the same steps, as we mentioned in above experiment. Nevertheless, we employed fastText instead of word2vec in combination with ABAE model. This experiment similarly applied orthogonal Procrustes and CCA for aligning pre-trained fastText word vectors with vectors learned on our corpus. As a consequence, we obtained the effective 300 dimensional word vectors, as we did above. 300 dimensional pre-trained fastText word vectors were taken by utilizing chaikin downloader. These pre-trained vectors were trained on Wikipedia data.

Once the alignment process is complete, we invoked ABAE model with the 300 dimensional aligned word vectors. This was followed by generating aspects and measuring the average coherence scores for each aspect size by employing the usual word vector similarity with the help of pre-trained fastText vectors, as mentioned

above. This experiment was carried out for aspect sizes starting from 7 to 50 to find the optimal number of aspects within the corpus.

Results and Discussions

Figure 6.10 depicts that average coherence score for aspect, which was generated by ABAE model with pre-trained fastText vectors, followed by orthogonal Procrustes method for aligning the pre-trained embedding space is very similar to that of ABAE model with pre-trained fastText vectors, followed by CCA method. This explains that both the alignment process, orthogonal Procrustes and CCA methods got along with pre-trained fastText vectors and ABAE model. However, since orthogonal Procrustes aligned word vectors with ABAE model gave maximum average coherence score at aspect size 9, so we chose orthogonal Procrustes aligned word vectors for ABAE model. Comparing the figure 6.6 with this one, we can conclude that pre-trained fastText word vectors followed by alignment process gave better coherence score than the fastText vectors trained on the corpus. This is true for all the aspects within the range. So, we can state that ABAE model with the help of alignment process of pre-trained fastText vectors could generate higher quality of aspects compared to that of fastText vectors trained on the corpus.

Overall Results and Discussions

We have discussed so far how we adapted the pre-trained vectors generated by different models, such as GloVe, word2vec and fastText to our corpus. We also

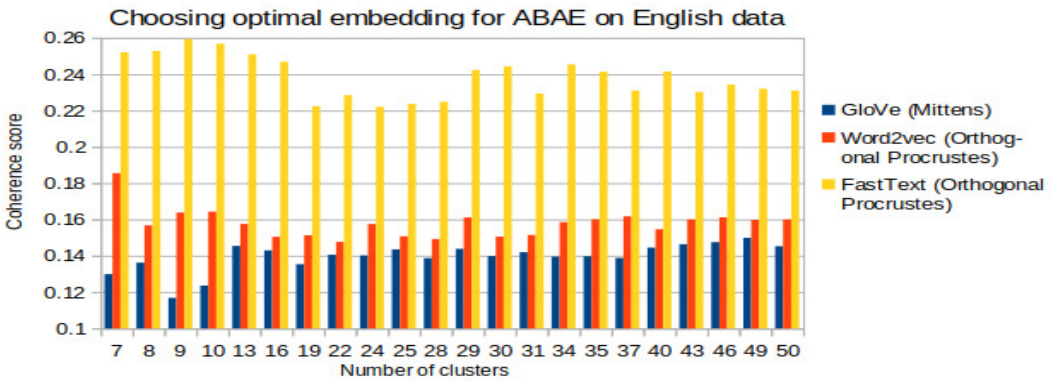


Fig. 6.11.. This figure displays average coherence score of ABAE model applied with different embeddings on English organic dataset for different aspect sizes between 7 and 50. The yellow bars and red bars represent the scores of the aspects generated by ABAE model with fastText and word2vec vectors using orthogonal Procrustes on pre-trained ones respectively. The blue bars are the corresponding scores using fine-tuning of pre-trained GloVe embedding.

elaborated how experiments connected these models to ABAE and corresponding results. Since we were interested in knowing which embedding model would work better with ABAE for extracting better quality of aspects, figure 6.11 portrays a comparative study to understand which pre-trained embedding model followed by either fine-tuning or embedding space alignment got along with ABAE model for better clustering of aspect terms. We noticed that ABAE with pre-trained fastText vectors using orthogonal Procrustes produced maximum coherence score than the others. This observation was even valid for all the aspect sizes within the range between 7 and 50. So we can conclude that fastText-based ABAE model generated higher quality of aspect out of the corpus using orthogonal Procrustes on pre-trained fastText vectors.

Furthermore, we can also conclude from the scores, which is displayed in the figure that application of orthogonal Procrustes over pre-trained word2vec vectors generated better quality of aspects than the fine-tuning of GloVe pre-trained vectors, when both of them were applied with ABAE model on our dataset. These last two conclusions were also observed in the set of experiments, which we performed in section 6.2.1.

6.2.3 Optimal Word Embedding Applicable with ABAE on English Data

In the section 6.2, we elaborated all the experiments we performed with ABAE model on English data for clustering aspect terms. This thesis also explored and discussed

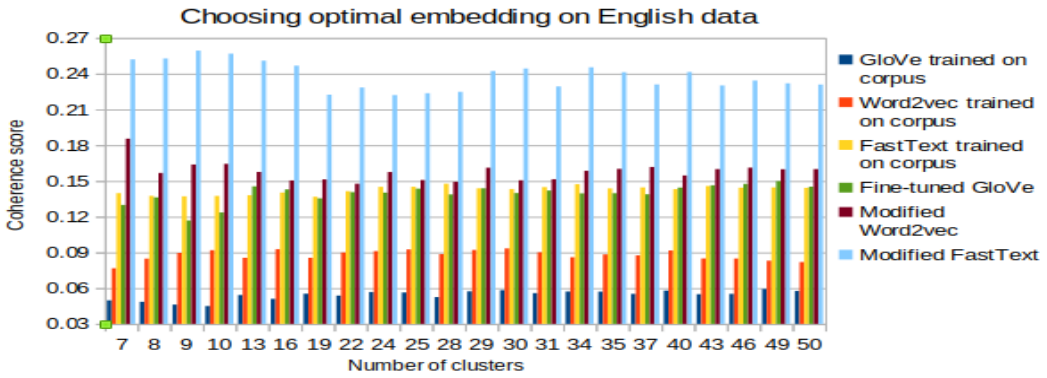


Fig. 6.12.. This figure outlines the average coherence score of all the aspects within the range between 7 and 50, generated by ABAE model with different embeddings. The sky blue bars and violet bars represent the scores, caused by ABAE model with pre-trained fastText vectors and pre-trained word2vec, which are followed by orthogonal Procrustes respectively. Similarly green bars are for ABAE model with fine-tuned GloVe vectors. The yellow bars, red bars and blue bars represent the scores, caused by ABAE model with fastText vectors, word2vec vectors and GloVe trained on the corpus respectively.

impacts of training GloVe, word2vec and fastText model on our corpus (see section 6.2.1) as well as fine-tuning or adapting pre-trained vectors of these model to our corpus (see section 6.2.2). Therefore, we were inquisitive to understand which embedding out of all the embeddings and their variants (e.g. pre-trained followed by fine-tuning) worked better with ABAE model in clustering aspect terms of English organic food data.

Figure 6.12 illustrates the results and outcomes of all the experiments together, which we conducted so far for identifying which embedding helped ABAE model for extracting better words in English data. Having observed the average coherence score for each aspect size ranging from 7 to 50 for all six experiments, we can deduce that pre-trained fastText model, followed by orthogonal Procrustes was efficient and effective for ABAE model, as its coherence score achieved maximum score for all the aspect sizes. Therefore, we can conclude that ABAE model with pre-trained fastText model with the help of embedding space alignment process generated higher quality of aspects for English organic food data.

6.3 Aspect Extraction on German Data

As we outlined in section 6.2, MUSE framework requires word vectors for the words in German data as well, so that MUSE would perform embedding space alignment between English and German, which is beneficial for bilingual aspect extraction. This motivated us to find optimal embedding, which would get along with ABAE model on German data, as we did on the English data. This explains why this thesis explored different embedding models along with ABAE model by performing a set of experiments on German data.

6.3.1 Word Embedding Trained on Organic Data

This section elaborates a set of experiments on ABAE model with different embeddings, such as GloVe, word2vec and fastText trained on our German corpus, as we conducted on English corpus (see in section 6.2.1).

GloVe

This experiment focused on how ABAE model would perform on the German organic food data for extracting semantically similar words with the help of GloVe model trained on our German corpus. For that reason, we trained GloVe model on our

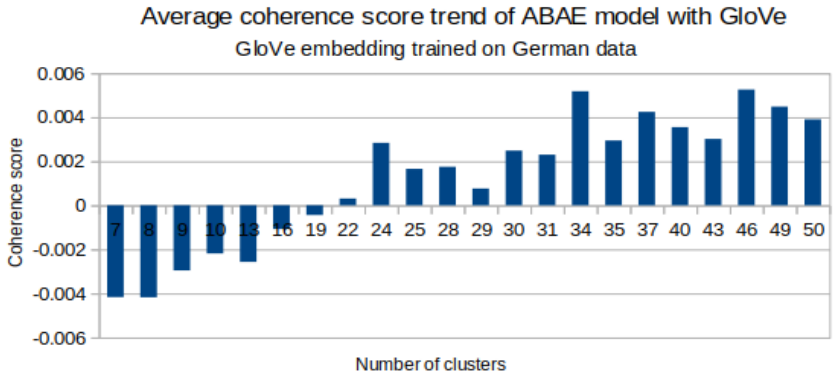


Fig. 6.13.. This figure illustrates the average coherence scores for all the aspects in the range between 7 and 50. These aspects were generated by ABAE model with GloVe embedding trained on the German data.

corpus using the same implementation and same hyperparameter setup (see table 6.3a), as we did on our English corpus (see experiment 6.2.1). Thus we obtained 300 dimensional vector representation of each word in the vocabulary of German corpus. Since we chose 300 dimensional vectors for word representation on English data, so to maintain the parity across experiments between English and German, we also chose word vectors to be 300 here.

Post learning word representations, ABAE model was invoked on the German

Hyperparameters	Values
VOCAB_SIZE	19000
LEARNING_RATE	0.001
EPOCHS	15
BATCH_SIZE	64
NEGATIVE_SAMPLES	10
REGULARIZATION_WEIGHT	0.1
MAX_LENGTH	100

Tab. 6.6.. This table outlines a set of hyperparameters for ABAE model for the experiments on German data.

corpus with the hyperparameter setting (see table 6.4), as we used for English data. However, we changed hyperparameter MAX_LENGTH to 100 instead of 150, since we noticed in German data that sentences having words around 100 was dominant (see appendix A.1.3). Table 6.6 shows hyperparameters' setup, which we used for experiments on German data. ABAE model then using 300 dimensional word representation, clustered the aspect terms. After that, we calculated the average coherence scores for each cluster size using word vector similarity (see section 3.7.2) to measure to what extent words were similar within a cluster. Since this task required us to provide pre-trained GloVe vectors for calculating vector similarity, we

utilized 300 dimensional pre-trained vectors provided by deepset company[15]. We conducted this experiment for aspect sizes ranging from 7 to 50 to figure out optimal number of aspects in the corpus.

Results and Discussions

Figure 6.13 depicts the average coherence scores for all aspect sizes ranging from 7 to 50, generated by the above experiment. We observe in the figure that this experiment yielded very low coherence scores for the aspect sizes between 7 and 22, in fact the scores were negative. This implies that these clusters of aspect terms no way contained semantically similar words. Coherence score of aspect sizes 34 and 46 were very comparable, though ABAE model generated maximum score at aspect size 46. So, we conclude that there were 46 optimal aspects in the German data.

Word2vec

In this experiment, ABAE model was explored along with word2vec embedding to observe how training word2vec model on the German corpus would affect ABAE model in extracting better quality of aspects. For that reason, we trained the word2vec model first on the German data with the necessary hyperparameter setup (see table 6.5) to learn the 300 dimensional word vectors.

Once we learned the word representation, ABAE model was executed with those word vectors and corresponding hyperparameter settings (see table 6.6), as we conducted above. ABAE as a part of the process, generated clusters of aspect words for different cluster sizes. This experiment was conducted for cluster sizes in the range between 7 and 50. Since we were interested in knowing to what extent cluster of aspect words cohere, we calculated average coherence score for the cluster sizes using word vector similarity, as we utilized in above experiment. For this reason, we utilized 300 dimensional pre-trained word2vec vectors[16] trained on German Wikipedia, provided by Yamada et al. (2018).

[15] deepset GmbH provides pre-trained GloVe vectors and it is available in `https://deepset.ai/german-word-embeddings`

[16] Word2vec vectors of 300 dimensions, which is pre-trained on German Wikipedia, is available on `https://wikipedia2vec.github.io/wikipedia2vec/pretrained/`

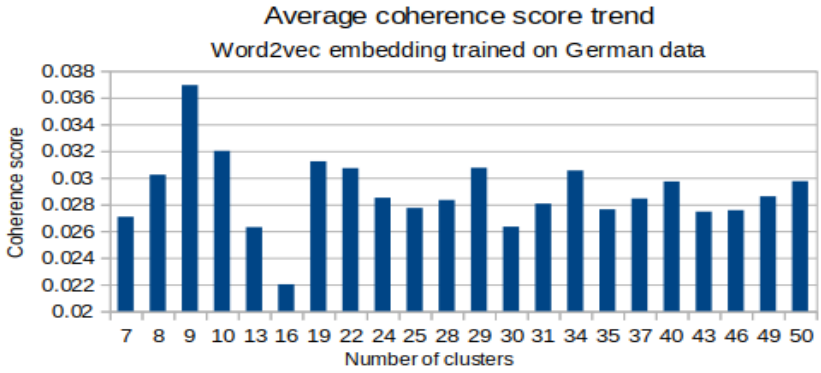


Fig. 6.14.. This figure depicts the average coherence scores for all the aspects ranging from 7 to 50. These aspects were generated by ABAE model with word2vec embedding trained on the German data.

Results and Discussions

Figure 6.14 reflects how efficient ABAE model with word2vec trained on our German corpus is in extracting high quality of aspects. Since this figure depicts the average coherence score for all the aspect sizes lying in the range between 7 and 50, we can state that the model gave maximum score at aspect size 9 and gave worst score at aspect size 16. So, we can conclude that ABAE model with word2vec embedding model identified 9 higher quality of aspects out of the corpus, when word2vec was trained on the corpus.

FastText

This experiment explored fastText embedding model with ABAE clustering model for the purpose of clustering coherent aspect terms. So, at the very beginning, we trained fastText model on the German corpus to learn 300 dimensional word vectors. Since training fastText model requires essential hyperparameters' values to be provided, we utilized the same hyperparameter (see experiment 6.2.1), as we employed while conducting experiments on the English corpus.

Once we learned 300 dimensional fastText vectors from our corpus, we employed ABAE model using those. ABAE model utilized the hyperparameters' setting (see table 6.6), as we employed in experiments for extracting aspects from German data. This was followed by measuring to what extent words in the resultant cluster are coherent. So, this experiment utilized word vector similarity (using equations 3.33) for this sub-task, like other experiments. Thus, this sub-task necessitated us to utilize chakin library to obtain 300 dimensional pre-trained fastText vectors. These vectors were trained on Wikipedia. chakin library is a downloader for pre-trained word

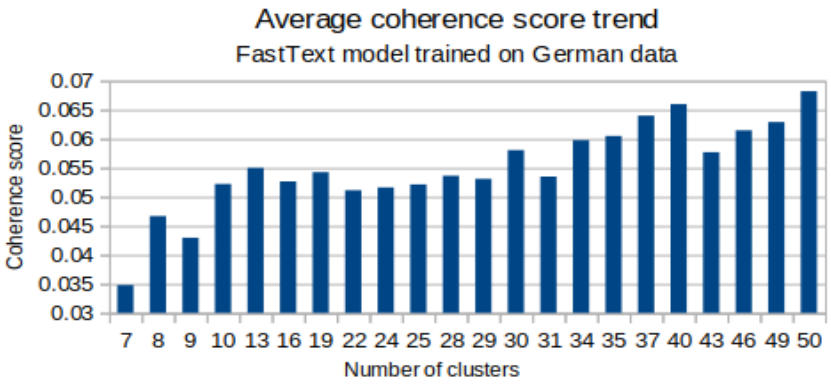


Fig. 6.15.. This figure depicts the average coherence scores for all the aspects in the range between 7 and 50. These aspects were generated by ABAE model with fastText embedding trained on the German data.

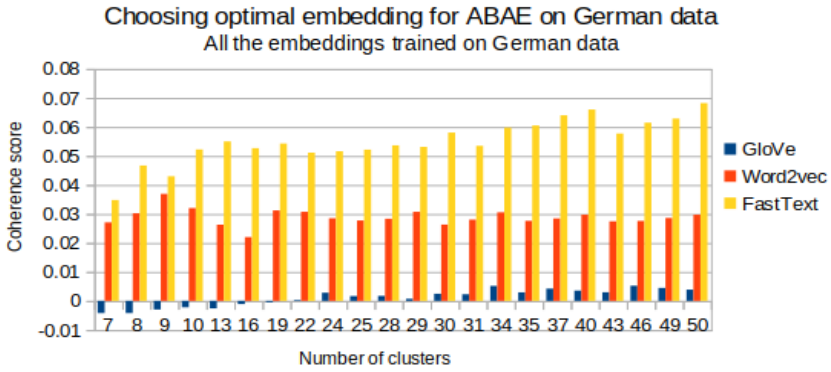


Fig. 6.16.. This figure outlines the average coherence scores for all the aspects in the range between 7 and 50. These aspects were generated by ABAE model with different embeddings trained on the German data. The blue bars, red bars and yellow bars denote the scores generated by ABAE model with GloVe, word2vec and fastText respectively.

vectors, as mentioned earlier. We repeated this experiment for the cluster sizes in the range between 7 and 50 to observe which one would be optimal aspects within the corpus using ABAE with fastText model.

Results and Discussions

Figure 6.15 portrays the average coherence score trend for all the cluster sizes ranging from 7 to 50, produced by above experiment. We notice in the figure that there were continuous jitters in the score. However. coherence score generated for aspect size 50 achieved maximum value out of all the scores in the range. So, we infer that according to ABAE model with fastText our German corpus had 50 aspects as the optimal size.

Overall Results and Discussions

We have discussed so far the impacts of different embedding models, such as GloVe, word2vec and fastText in association with ABAE, while they were trained on our German corpus. However, we were motivated in knowing which embedding model would work better with ABAE for the purpose of extracting higher quality of aspects on German corpus. On that account, figure 6.16 represents the collective results of above three experiments. We observe in the figure that results of ABAE model with fastText trained on German corpus dominated over other two experiments with GloVe and word2vec-based ABAE model in terms of coherence scores. So, we can conclude that ABAE model with fastText trained on German data produced higher quality of aspects that other two models. Furthermore, it is also noticeable in the figure that GloVe-based ABAE model generated significantly low coherence score compared to other two models. This explains that ABAE model with GloVe trained on German data was not at all suitable on our German data.

6.3.2 Fine-tuned Word Embedding and Embedding Space Alignment

Our German dataset was small in number of sentences like our English dataset. It contained 40620 sentences. As we explained in section 6.2.2, since embedding models require lot of parameters to be updated, training these embedding models on small volume of data is not worthwhile. On that account, we utilized pre-trained embeddings followed by either fine-tuning or aligning embedding space onto our corpus, like we did on English data (see section 6.2.2).

GloVe

This experiment explored how ABAE model performed on our corpus with fine-tuned GloVe embedding. At first, we collected 300 dimensional pre-trained GloVe vectors, trained on German Wikipedia. As mentioned earlier in experiment 6.3.1, company deepset GmbH has open-sourced these pre-trained embeddings. Furthermore, we fine-tuned the pre-trained vectors the way we did on the English corpus in experiment 6.2.2 using Mittens.

Post fine-tuning, once we learned the effective and efficient 300 dimensional word representation, this experiment invoked ABAE model on the German corpus utilizing the representations and the same hyperparameters' settings (see table 6.6) we used so far. As a result, ABAE model clustered aspect terms out of the corpus. This

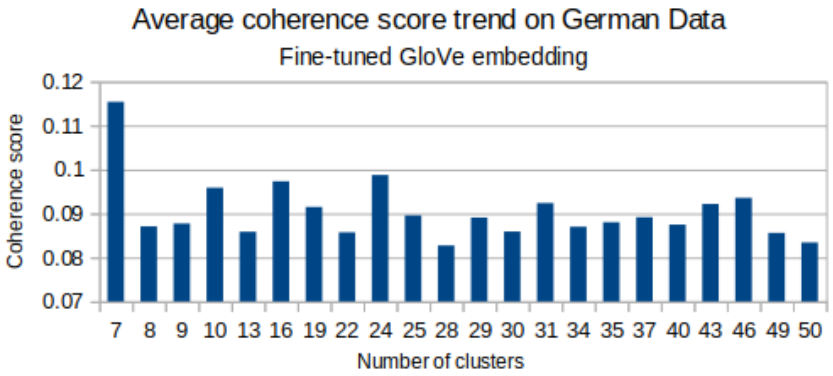


Fig. 6.17.. This figure illustrates the average coherence scores for all the aspect sizes ranging from 7 to 50. These were generated by ABAE model with fine-tuned GloVe vectors.

execution was repeated for the cluster sizes ranging from 7 to 50. In addition, as we conducted several times, we calculated the average coherence score for each cluster size in the said range between 7 and 50 using word vector similarity. For this reason, we again utilized pre-trained 300 dimensional GloVe vectors.

Results and Discussions

Figure 6.17 reflects that the average coherence scores for aspect sizes between 8 and 50, generated by the above experiment were close to each other. On that account, we can deduce that quality of aspects for aspect sizes between 8 to 50 were similar to a large extent. However, coherence score for aspect size 7 was found to overshoot the rest of the aspects in the range to a large extent. So, we can conclude that this experiment generated higher quality of aspects for aspect size 7. This concludes that according to ABAE with fine-tuned GloVe, our German corpus had 7 optimal aspects.

Comparing this result with result of experiment of ABAE model with GloVe trained on German data (see figure 6.13) depicts the significant improvement, which we achieved by using pre-trained vectors followed by fine-tuning.

Word2vec

In this experiment, ABAE model was explored with pre-trained word2vec vectors followed by some embedding space alignment methods to analyze to what extent this approach would generate meaningful clusters from the German data. At the very first step, we obtained 300 dimensional word2vec vectors using Wikipedia2vec tool provided by Yamada et al. (2018). These vectors were trained on German Wikipedia.

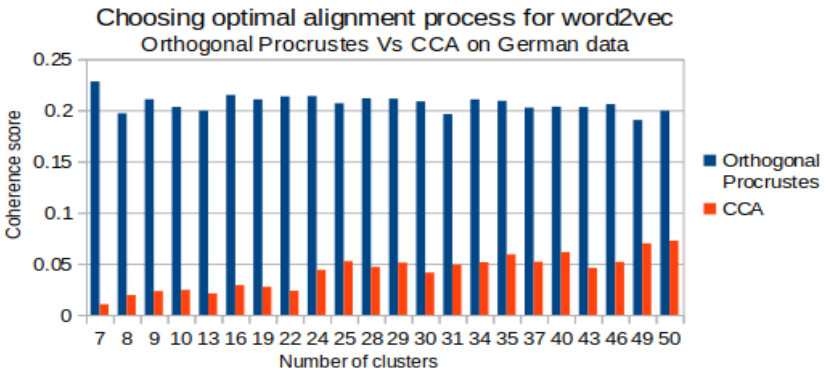


Fig. 6.18.. This figure reflects the average coherence scores for all the aspect sizes ranging from 7 to 50. These were generated by ABAE model with pre-trained word2vec vectors, followed by embedding space alignment process. The blue bars and red bards denote the scores when orthogonal Procrustes and CCA applied on pre-trained word2vec vectors respectively.

As we conducted experiment 6.2.2 on English data, here also we employed both the orthogonal Procrustes and CCA methods for aligning the pre-trained vectors' space with our German corpus to observe which method would perform better with ABAE model.

Once we obtained the aligned 300 dimensional word vectors by both the methods, ABAE model was invoked with those vectors for extracting aspect terms and clustering them. We utilized same hyperparameters' configurations (see table 6.6) for these. Post which, in order to measure the average coherence for each aspect size, we employed word vector similarity metric. This sub-task required us to provide 300 dimensional word2vec pre-trained word vectors, which we utilized above. This experiment was executed for different aspect sizes ranging from 7 to 50.

Results and Discussions

Figure 6.18 portrays that ABAE model with pre-trained word2vec vectors, followed by orthogonal Procrustes, generated more or less equivalent coherence score for each aspect sizes ranging from 7 to 50. However, CCA method over pre-trained word2vec vectors with ABAE model showed more or less increasing trend over average coherence scores, as we proceeded over a range of aspect sizes between 7 and 50. Nevertheless, orthogonal Procrustes dominated over CCA for all the aspect sizes in the said range on our German corpus. Thus, we can conclude that ABAE model performed better when we utilized orthogonal Procrustes over pre-trained word2vec compared to CCA. ABAE model with pre-trained word2vec vectors, followed by orthogonal Procrustes gave maximum score at aspect size 7. Thus, we can state that our German corpus had 7 optimal aspects.

FastText

We have explored so far pre-trained GloVe, word2vec either by fine-tuning or aligning embedding space along with ABAE model on German data. So, this experiment observed how pre-trained fastText, followed by embedding space alignment helped ABAE model for extracting high quality of aspects. We first collected 300 dimensional pre-trained fastText vectors using chakin downloader, as we did earlier. Then, like above experiment, we applied orthogonal Procrustes and CCA both on the pre-trained vectors to align the pre-trained embedding space onto our German corpus.

Post learning the aligned 300 dimensional vectors using both the methods, we triggered ABAE model on the German corpus with those vectors to cluster extracted aspect terms. After that, we calculated average coherence score for each aspect size to draw a conclusion about which embedding space alignment performed better. Since this score was calculated by using word vector similarity, as we did earlier, we again employed 300 dimensional pre-trained fastText vectors. This experiment was repeated several times for the aspect sizes ranging from 7 to 50.

Results and Discussions

Figure 6.19 reflects the average coherence scores for each aspect size in the range between 7 and 50 for both the methods, orthogonal Procrustes and CCA. We observe in the figure that ABAE generated higher scores for all the aspects in the range when pre-trained fastText model was aligned by orthogonal Procrustes than that of CCA. Thus, we can infer that like word2vec, embedding space alignment of orthogonal Procrustes worked better than CCA. Furthermore, ABAE model with pre-trained vectors, followed by orthogonal Procrustes achieved maximum score at aspect size 13. Thus, according to this method, our German document had 13 optimal aspects. Comparing this result with the result (see figure 6.15) we obtained in the experiment of ABAE model with fastText trained on our data, we can infer that utilizing fastText followed by orthogonal Procrustes improved significantly ABAE's performance.

Overall Results and Discussions

We have discussed so far impacts of different pre-trained embeddings followed by either fine-tuning or embedding space alignment on the performance of ABAE model. However, we were interested in knowing which embedding worked best with ABAE model for extracting higher quality of aspects. For this reason, figure 6.20 represents

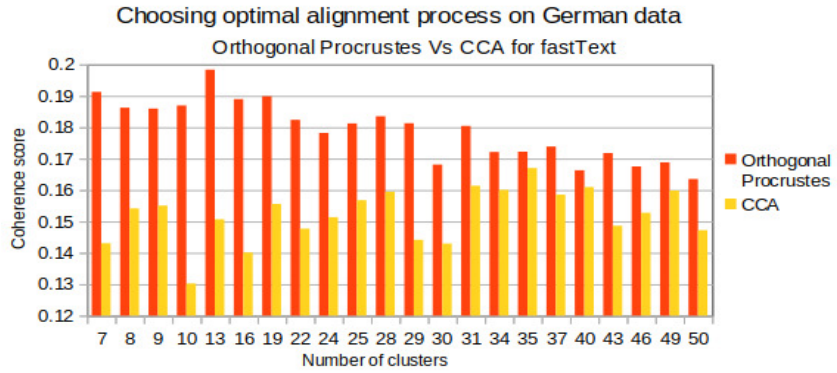


Fig. 6.19.. This figure outlines the average coherence scores for all the aspect sizes ranging from 7 to 50. These were generated by ABAE model with pre-trained fastText vectors, followed by embedding space alignment process. The red bars and yellow bards denote the scores when orthogonal Procrustes and CCA applied on pre-trained fastText vectors respectively.

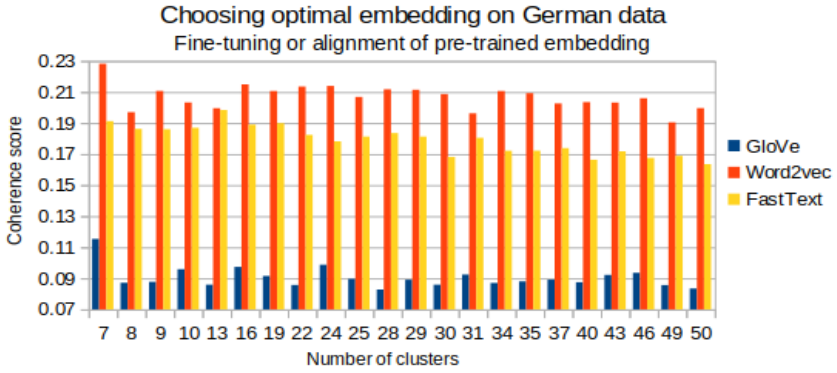


Fig. 6.20.. This figure reflects the average coherence scores for all the aspect sizes ranging from 7 to 50. These were generated by ABAE model with different pre-trained vectors, followed by embedding space alignment process or fine-tuning. The red bars and yellow bards denote the scores when orthogonal Procrustes applied on pre-trained word2vec and fastText vectors respectively. The blue bars denote the scores for fine-tuned GloVe.

the collective results of above three experiments. We see in the figure that ABAE model with pre-trained word2vec followed by orthogonal Procrustes gave better coherence scores than others. Thus, we can conclude that this one generated higher quality of aspects compared to other two experiments. Furthermore, we notice in the figure that fine-tuning of GloVe along with ABAE produced very low coherence scores. So we can deduce that fine-tuning of GloVe for ABAE model did not suit on German corpus.

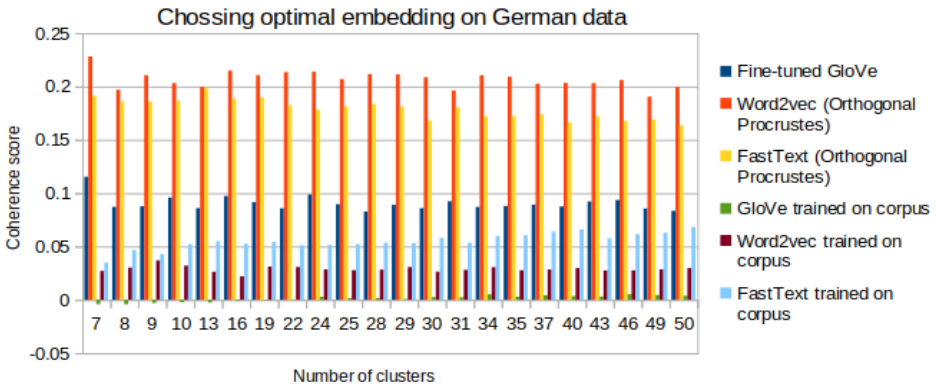


Fig. 6.21.. This figure outlines the average coherence scores for each aspect size ranging from 7 to 50. It is a collective results of all the experiments we conducted on German data altogether.

6.3.3 Optimal Word Embedding Applicable with ABAE on German data

Figure 6.21 displays a comparative study of all the experiments we conducted on German data. While observing the coherence score in the figure, it can be noted that ABAE model with pre-trained vectors followed by either fine-tuning or embedding space alignment always generated better score than the ABAE model with embedding method trained on our corpus. Thus, we can conclude that utilizing pre-trained vectors and then adapting them to the corpus was an effective and efficient solution.

More over, we observed that ABAE model achieved maximum score whenever pre-trained word2vec vectors were aligned by orthogonal Procrustes. Thus, orthogonal Procrustes over word2vec model played an important role for ABAE model to extract higher quality of aspects.

6.4 Bilingual Aspect Extraction

All the experiments we conducted till now, were associated with extracting aspects on English and German corpus separately and finding the optimal embedding model, which performed better with ABAE on these corpora separately. However, as the this thesis's title portrays, we were more into multilingual corpora, which were in our case English and German together. On that account, we combined English and German corpus together and formed a bilingual corpora for our next course of tasks regarding aspect extraction.

We so far noted optimal embedding model on English corpus and similarly on German corpus. At this stage, we did not have any bilingual embedding model,

which would work with ABAE model on the corpora. Furthermore, embedding spaces of these embedding model were also different, so simple concatenation of these two embedding spaces / matrices would not be sufficient for building bilingual embedding model. This necessitated us to employ MUSE[17] (discussed in section 3.1.4) to provide a mapping between optimal embedding model of English and German corpus. Thus MUSE helped us to align optimal embedding of English data to embedding space of German data; leading to bilingual embedding space.

The thesis observed that ABAE model with pre-trained fastText embedding, followed by orthogonal Procrustes method performed better than other embeddings on our English corpus (discussed in section 6.2.3), whereas ABAE model with pre-trained word2vec embedding, followed by orthogonal Procrustes method performed better than other embeddings on our German corpus (discussed in section 6.3.3). Hartmann et al. (2018) showed that MUSE cannot perform well if we provide two embedding spaces, generated by different algorithms. For that reason, we did not invoke MUSE using these embedding spaces, instead we conducted two experiments with ABAE. One was invoking MUSE with optimal word2vec vectors on English corpus as source space and optimal word2vec vectors on German corpus as target space. This was followed by applying ABAE utilizing the generated bilingual embedding space. Another experiment was using MUSE with optimal fastText vectors on English corpus as source space and optimal fastText vectors on German corpus as target space. This was also eventually followed by ABAE. Therefore, in the next sections we outline these two experiments.

6.4.1 Word2vec

In this experiment, we first invoked MUSE on word2vec vector space, generated by orthogonal Procrustes method over pre-trained vectors for English data, so that MUSE would align this space to the embedding space of vectors, obtained by applying orthogonal Procrustes method over pre-trained word2vec vectors for German data. MUSE follows two ways: supervised and unsupervised. We utilized supervised MUSE for alignment. Furthermore, MUSE needs certain arguments for operation. We utilized same arguments and values, provided by the github repository[18] of MUSE. Since post this process English word2vec embedding was aligned with German word2vec embedding, so concatenating these two embedding matrices resulted in our bilingual embedding vectors.

ABAE model was invoked then with these embedding vectors on our bilingual corpora to extract aspect terms and cluster them. Since ABAE requires certain hy-

[17] Code of MUSE is available in `https://github.com/facebookresearch/MUSE`
[18] Github repository of MUSE is `https://github.com/facebookresearch/MUSE`

perparameters to be provided, we utilized same hyperparameter configuration (see table 6.4), as we employed earlier experiments. Once ABAE model generated the cluster of aspect terms, we calculated average coherence score using word vector similarity (see section 3.7.2) as we used in earlier experiments. For this task, we used our MUSE over word2vec based bilingual word vectors. Furthermore, we also computed coherence score using UMass coherence score (see section 3.7.1) to observe to what extent clusters represent the bilingual corpora. This experiment was also executed for aspect sizes ranging from 7 to 50.

6.4.2 FastText

This experiment is very similar to the above experiment. Here, at the very first step, instead of word2vec, MUSE was applied with fastText model. Embedding space generated by applying orthogonal Procrustes method over pre-trained fastText vectors for English data was provided to MUSE as source. Similarly, embedding space generated by applying orthogonal Procrustes method over pre-trained fastText vectors for German data was provided to MUSE as target. MUSE then aligned English fastText embedding space to German fastText embedding space. Post that, as we did above, we formed bilingual word embedding by concatenating German fastText embedding matrix and aligned English fasText embedding.

Once it was done, we employed ABAE model with the bilingual embedding and hyperparameters' configuration (see table 6.4), as discussed above. After ABAE generated clusters of aspect terms, we calculated to what extent words within a cluster were coherent by utilizing both the methods, word vector similarity and UMass coherence score, as we followed in above experiment. For the word vector similarity based evaluation metric, we used our MUSE over fastText based bilingual word vectors. We repeated this experiment for aspect size in the range between 7 and 50.

6.4.3 Results and Discussions

Figure 6.22 illustrates a comparative study between the two experiments (6.4.1 and 6.4.2) we mentioned above. While looking at the average coherence scores for each aspect in a range between 7 and 50 for both the experiments, we can state that MUSE with fastText most of times dominated over MUSE with word2vec. We also noticed that ABAE with MUSE gave maximum score at aspect size 22, when MUSE was applied with fastText. Therefore, using this evaluation metric, we can conclude that our bilingual data had 22 optimal topics. Furthermore, we have seen that MUSE with word2vec based ABAE generated very lower score compared to MUSE with

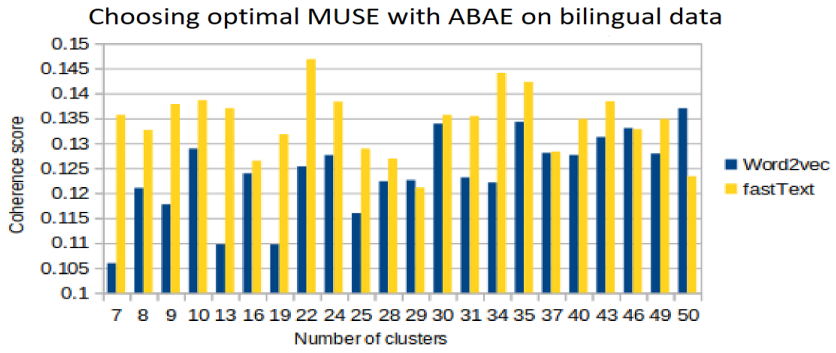


Fig. 6.22.. This figure depicts the average coherence scores (calculated using word vector similarity) for each aspect size ranging from 7 to 50. These aspects were generated by ABAE model with bilingual embedding, generated by MUSE. The yellow denote the scores, when MUSE was fed with pre-trained fastText vectors, followed by orthogonal Procrustes for both English and German. Similarly the blue bars denote the scores for same experiment setup. However, instead of fastText, word2vec was used there.

fastText for aspect sizes 7, 13 and 19.

Figure 6.23 also reflects same comparison. However, here we used UMass coherence score for measuring average coherence scores. We observe in the figure that for small cluster sizes, such as 7,8,10 etc. MUSE with word2vec dominated over MUSE with fastText. As we increased number of clusters, average scores for MUSE with fastText increased over word2vec. Eventually, MUSE with fastText achieved maximum score at aspect size 46 within this range. Therefore, we can infer that MUSE with fastText worked better with ABAE as we granulated the topics.

Furthermore, we manually compared the clusters of words for both the aspect sizes

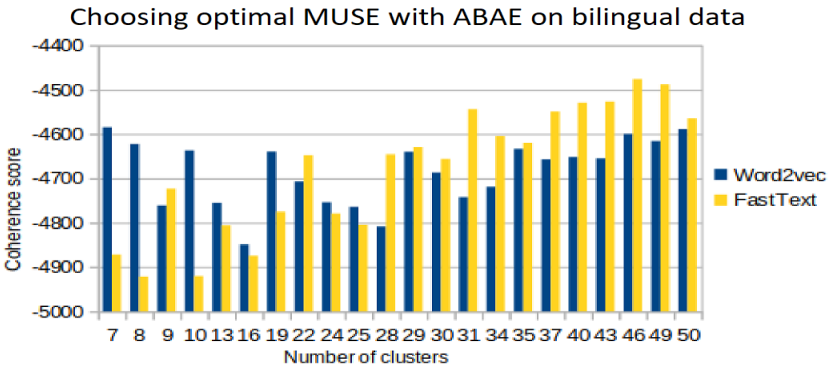


Fig. 6.23.. This figure illustrates the average coherence scores (calculated using UMass coherence score) for each aspect size ranging from 7 to 50. These aspects were generated by ABAE model with bilingual embedding, generated by MUSE. The yellow denote the scores, when MUSE was fed with pre-trained fastText vectors, followed by orthogonal Procrustes for both English and German. Similarly the blue bars denote the scores for same experiment setup. However, instead of fastText, word2vec was used there.

22 and 46, since both were selected as optimal cluster size by the two different evaluation metrics. We noticed that even though aspect size 46 had more general clusters as compared to that of 22, the aspect terms for 46 are very meticulous. The aspect terms for 22 on the other hand display lower level of coherence and are not deterministic enough. Additionally, since both the figures reflected that MUSE with fastText performed better with ABAE on our bilingual data, we can note that ABAE with MUSE over fastText is a model in our bilingual data.

For the clarity of understanding below table provides a summarization of all the experiments this thesis conducted for finding a suitable ABAE model with an aim towards multilingual aspect extraction:

Dataset	Embedding	Best aspect size	Best coherence score
English	GloVe	49	0.15
	Word2vec	7	0.18
	FastText	9	**0.26**
German	GloVe	7	0.12
	Word2vec	7	**0.23**
	FastText	13	0.20
Bilingual	MUSE with word2vec	7	-4584.98
	MUSE with fastText	46	**-4476.42**

Tab. 6.7.. This table illustrates optimal aspects and optimal coherence score (approximated to two decimal points) achieved using different embedding models with ABAE on different corpus. Bold figures represent the best models out of all the models on the corresponding corpus. Scores on bilingual data were measured by UMass coherence score, whereas word vector similarity was applied on English and German data.

6.4.4 Aspect Distributions over Bilingual Data

This thesis elaborated how multilingual aspect extraction can be achieved and explored different parameters and optimized algorithm for the task. One important task this thesis explored was to determine the aspects that bilingual organic food data consisted of and how broadly they were discussed in our social media data. In other words, this work also shows the distribution of topics or aspects in order to figure out what the population is concerned about or their opinions in general based on the comments in both English and German.

Appendix A.3 depicts such a distribution for both German and English data combined. It can be observed that a large percentage of the comments deal with 'Gastronomy' indicating that most of the comments revolve around food related topic, closely followed by 'Farming'. Other topics like 'Supermarket', 'Complementary adjectives',

'Organization' all related to organic food or food in general occupy the top positions as trending topics. The inferred labels or topics from the aspect terms within each cluster generated by our ABAE model for the combined dataset in given in appendix A.2.

7 Conclusion and Future Work

This thesis was initiated with the motivation for extracting aspects from multilingual corpora, such that representative words of an aspect are semantically similar. So we selected ABAE model over the data. Hence, we performed several experiments on English and German corpus to figure out which embedding model would produce higher quality of aspects using ABAE on the data. We observed that ABAE with fastText and ABAE with word2vec produced good quality of aspects on English and German data respectively. These results helped us apply MUSE, which was followed by ABAE model on the bilingual corpora. Finally, from the results we concluded that ABAE with MUSE over fastText would give better quality of aspects on the bilingual corpora.

In this section, based on our findings throughout the thesis, we will address some specific research questions.

- **Which number of clusters are sufficient and reasonable for the domain of interest?**
  We observed in the outcomes of all the experiments on our organic data that there were no specific optimal cluster size. Based on the embedding algorithms, ABAE model changed the optimal number of aspects in the corpora. Despite this fact, we decided with the help of coherence scores of clusters that ABAE model with MUSE over fastText generated best optimal topics over the bilingual data, which was 46.

- **Which degree of granularity is sufficient?**
  We observed throughout the thesis that, with lower number of clusters, the aspects generated by ABAE model tends to have multiple topics categorized into one cluster. However, with increased cluster sizes, more coherent and fine-grained clusters are generated. Consequently, words within the fine-grained clusters are relatively much more deterministic in assessing the overall topic or aspect of the cluster. Therefore, we maintained a trade-off in identifying optimal cluster size by using our own perception and coherence score metrics.

- **When are the cluster more or less coherent?**
  As we stated above, since fine-grained clusters granulate the topics over the

corpus, we get clusters of more coherent words to represent fine-grained topics. However, since coarse-grained cluster consists of multiple topics, it contains much more dissimilar words compared to the other case. So it generates less coherence score. As we see in the figure 6.23, aspect sizes 8,10 and 16, generated by ABAE with MUSE over fastText produces less coherence compared to clusters in higher range, especially between 37 and 50.

- **How do these factors depend on the chosen word embeddings?**
  Since different embeddings are built upon different algorithms, we experienced throughout the course of this work that it changes clustering of coherent words significantly. As we see in the figure 6.23, coherence score trend is different for MUSE with fastText and MUSE with word2vec. If we had chosen MUSE with word2vec, then aspect size achieving the maximum would be 7 as opppsed to 46 for MUSE with fastText. Therefore, we conducted several experiments to choose best embedding model on our organic data.

Although this thesis tried to explore several directions towards multilingual aspect extraction, such as finding optimal word embedding, noting how ABAE performs over multilingual data, optimizing few hyparameters and different coherence score evaluation metrics, due to scarcity of time we could not explore all the possible directions. There are some tasks left unexplored. One potential task could be exploring other coherence score evaluation metrics as well. Röder et al. (2015) provides a framework for measuring several coherence scores. In future, this could be used to measure the coherence with respect to several evaluation criteria instead of one or two criteria to estimate the performance of the model. In addition, more hyperparameters of the model can be optimized. We only used MUSE supervised method for alignment. Additionally, MUSE unsupervised could be explored. In this thesis, we mapped English embedding space to German embedding space, reverse direction can be explored in future.

This thesis explored English and German data as a part of intended task. However, it is worthwhile to extend it for more than two languages in future. Furthermore, since our motivation was increasing coherence scores, a potential future task could be incorporating coherence score in the loss function of ABAE model. It would be interesting to observe how the model behaves. Another potential point is that though we preprocessed data and cleaned it, it is still not very structured and cleansed, since it is informal in nature. For example, we observed that all the spellings are not correct, more robust spelling correction dictionary is required. Therefore, we can employ more rules to clean the data further in future.

Appendix A

A.1 Dataset

A.1.1 An Example of English Unbiased Data of Our Dataset


```
{"article_title":"Organic food",
   "article_author":[
      {
         "article_author_id":"369825",
         "article_author_name":"PayalSethi"
      }
   ],
   "article_time":"2017-11-28 12:36:00",
   "article_text":"Hi, how many of you are into organic
      food? ... ",
   "article_source":"cafemom",
   "comments":[
      {
         "comment_id":"post349826036",
         "comment_author":{
            "comment_author_id":"2622512",
            "comment_author_name":"newwifenmom"
         },
         "comment_time":"2017-11-28 12:49:00",
         "comment_text":"I think that if you can, organic
             and local is best"
      }
   ],
   "search_query":"organic food site:cafemom.com",
   "article_url":"http://www.cafemom.com/group/121506/
      forums/read/21760067/Organic_food",
   "resource_type":"forum",
   "relevant":1}
```

A.1.2 Dataset Statistics

Language	Type	Source Type	Total Comments	Relevant Comments
English	Biased	NA	306037	86728
	Unbiased	News sites	101711	26814
		Forums	107599	27394
German	Biased	Forums	374	314
		Blogs	5791	1637
	Unbiased	News sites	195429	92668
		Forums	25856	12841

Tab. A.1.. (Source: Halder, 2019) This table outlines forums, blogs and news articles wise count of social media comments for English and German organic food data. Here, "*NA*" stands for "*not applicable*". Since we were not certain of source of English biased data, we used "*NA*".

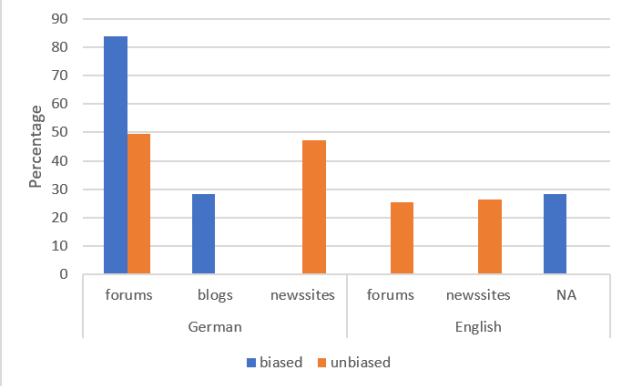


Fig. A.1.. (Source: Halder, 2019) This figure illustrates forums, blogs and news articles wise distribution of relevant comments over whole dataset for English and German data. As we mentioned in above table, "*NA*" stands for "*not applicable*" and represents biased English data. The blue and orange bars denote biased and unbiased source of data respectively.

A.1.3 Distribution of Sentences over Their Length on German Data

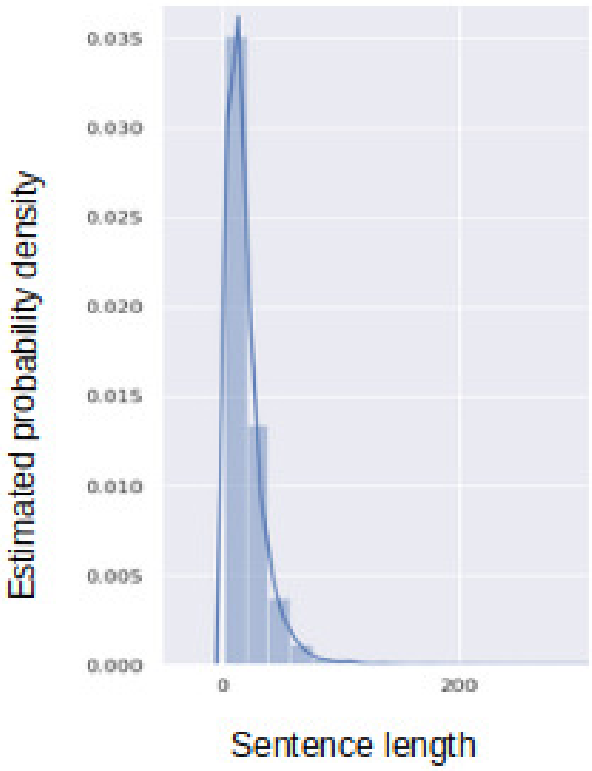


Fig. A.2.. This figure displays the probability density of the length of sentences of the German corpus. Density signifies the number of sentences with corresponding length.

A.2 Results

A.2.1 Aspect Distribution over Bilingual Data

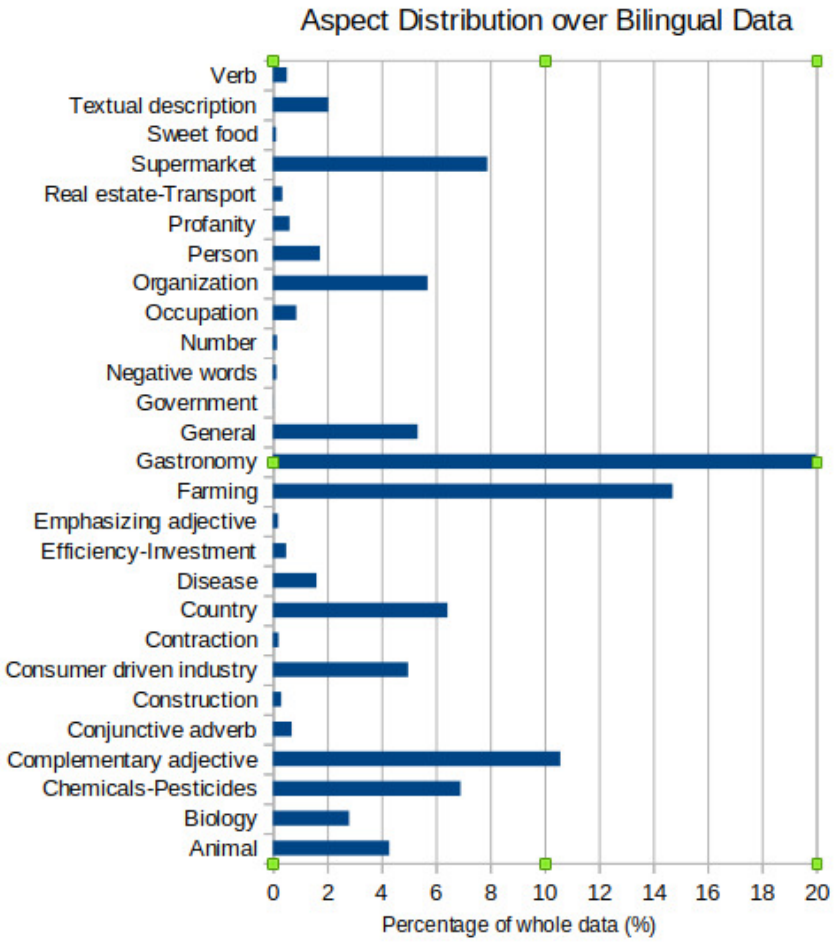


Fig. A.3.. This figure illustrates aspect wise distribution of sentences of bilingual data. For example, 20% of the data are related gastronomy and this is maximum as per the observation in the figure.

A.2.2 Aspect and Representative Aspect Terms over Bilingual Data

Inferred Labels	Aspect Terms
Gastronomy	lecker, salad, essen, pizza, gourmet, Schokolade, eating, dessert, isst, Käse, Joghurt, snack, delicious, Marmelade
Farming	farming, monocrop, Gemüseanbau, farmer, Bioanbau, farm, Maisanbau, agronomist, Biolandbau, monocropping, Biobetrieb
Complementary adjective	elegant, edel, genuine, superior, strong, robust, spiritual, organic, originell, neuartig
Supermarket	Bio, Rewe, Edeka, Bioladen, Tegut, Supermarket, Discounter, Biosupermarkt, Lidl, Supermarktketten, Greenpeace, Biomarkt
Chemicals-Pesticides	herbicide, pesticide, glyphosate, sulphites, pyrethrin, atrazine, dicamba, bisphenol, organophosphate, Chemikalie
Country	america, europe, usa, india, britain, mexico, germany, australia, russia, england, spain, european, france
Organization	mitsui, wfs, fao, kansa, bpa, ftc, ige, vandana, gmp, usda, ncr, shroff, wapo, ota
Consumer driven industry	Lebensmittelbranche, Automobilindustrie, verbraucherfreundlich, Endverbraucher, Lebensmittelindustrie
Animal	Tier, animal, Schwein, Rind, Schaf, Tierart, kuh, Nutztier, cow, Kalb, huhn, mammal
Biology	Biologie, Lebewesen, biological, biology, Bodenbakterien, biochemical, ecology, Bodenlebewesen, Bodenorganismen, Ökologie
General	rain, topfen, winden, fressen, rollen, hocken, Beet, swell, menschen, kunden, nachbar, lage, anforderungen, regel, farm, einstellung, biespiel

Tab. A.2.. This table shows few top most inferred aspects and representative aspect terms of bilingual data which were generated by ABAE model with MUSE for optimal aspect size 46.

Bibliography

Arora, Ananta, Chinmay Patil, and Stevina Correia (2015). „Opinion Mining: An Overview“. In: *International Journal of Advanced Research in Computer and Communication Engineering* (cit. on p. 3).

Asnani, Kavita and Jyoti Pawar (Dec. 2017). „Automatic Aspect Extraction using Lexical Semantic Knowledge in Code-Mixed Context“. In: *Procedia Computer Science* 112, pp. 693–702 (cit. on p. 6).

Asnani, Kavita and Jyoti D. Pawar (2016). „Use of Semantic Knowledge Base for Enhancement of Coherence of Code-mixed Topic-Based Aspect Clusters“. In: *ICON* (cit. on p. 11).

Athiwaratkun, Ben, Andrew Gordon Wilson, and Anima Anandkumar (2018). „Probabilistic FastText for Multi-Sense Word Embeddings“. In: *CoRR* abs/1806.02901. arXiv: 1806.02901 (cit. on p. 19).

Backyard, Text Mining (Nov. 2017). *Correcting Words using Python and NLTK*. URL: https://rustyonrampage.github.io/text-mining/2017/11/28/spelling-correction-with-python-and-nltk.html (cit. on p. 42).

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio (2015). „Neural Machine Translation by Jointly Learning to Align and Translate“. In: *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (cit. on pp. 3, 10, 24, 25).

Bendersky, Eli (Oct. 2016). *The Softmax function and its derivative*. URL: https://eli.thegreenplace.net/2016/the-softmax-function-and-its-derivative/ (cit. on p. 34).

Blei, David M., Andrew Y. Ng, and Michael I. Jordan (2002). „Latent Dirichlet Allocation“. In: *Advances in Neural Information Processing Systems 14*. Ed. by T. G. Dietterich, S. Becker, and Z. Ghahramani. MIT Press, pp. 601–608 (cit. on pp. 5, 31).

Bojanowski, Piotr, Edouard Grave, Armand Joulin, and Tomas Mikolov (2016). „Enriching Word Vectors with Subword Information“. In: *arXiv preprint arXiv:1607.04606* (cit. on pp. ix, 6, 12, 18, 53).

Borga, Magnus (Jan. 2001). *Canonical Correlation a Tutorial*. URL: https://www.cs.cmu.edu/~tom/10701_sp11/slides/CCA_tutorial.pdf (cit. on pp. 21, 22).

Bouma, Gerlof (Jan. 2009). „Normalized (Pointwise) Mutual Information in Collocation Extraction“. In: *Proceedings of the Biennial GSCL Conference 2009* (cit. on pp. 27, 28).

Boyd-Graber, Jordan and David M. Blei (2009). „Multilingual Topic Models for Unaligned Text“. In: *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*. UAI ’09. Montreal, Quebec, Canada: AUAI Press, pp. 75–82 (cit. on p. 11).

Britz, Denny (Jan. 2016). *Attention and Memory in Deep Learning and NLP*. URL: `http://www.wildml.com/2016/01/attention-and-memory-in-deep-learning-and-nlp/` (cit. on p. 25).

Brody, Samuel and Noemie Elhadad (2010). „An Unsupervised Aspect-sentiment Model for Online Reviews“. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. HLT ’10. Los Angeles, California: Association for Computational Linguistics, pp. 804–812 (cit. on pp. 4, 5).

Cho, KyungHyun, Bart van Merrienboer, Dzmitry Bahdanau, and Yoshua Bengio (2014). „On the Properties of Neural Machine Translation: Encoder-Decoder Approaches“. In: *CoRR* abs/1409.1259. arXiv: `1409.1259` (cit. on p. 24).

Conneau, Alexis, Guillaume Lample, Marc’Aurelio Ranzato, Ludovic Denoyer, and Hervé Jégou (2017). „Word Translation Without Parallel Data“. In: *arXiv preprint arXiv:1710.04087* (cit. on pp. ix, 6, 12, 19, 20).

Cory, Simon (Oct. 2018). *The orthogonal Procrustes problem*. URL: `https://simonensemble.github.io/2018-10-27-orthogonal-procrustes/` (cit. on p. 21).

Dhillon, Paramveer, Dean P Foster, and Lyle H. Ungar (2011). „Multi-View Learning of Word Embeddings via CCA“. In: *Advances in Neural Information Processing Systems 24*. Ed. by J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger. Curran Associates, Inc., pp. 199–207 (cit. on p. 22).

Ding, Ran, Ramesh Nallapati, and Bing Xiang (2018). „Coherence-Aware Neural Topic Modeling“. In: *CoRR* abs/1809.02687. arXiv: `1809.02687` (cit. on pp. 7, 28, 29).

Dingwall, Nicholas and Christopher Potts (2018). „Mittens: An Extension of GloVe for Learning Domain-Specialized Representations“. In: *CoRR* abs/1803.09901. arXiv: `1803.09901` (cit. on pp. 36, 37).

Dragoni, Mauro, Marco Federici, and Andi Rexha (May 2018). „An unsupervised aspect extraction strategy for monitoring real-time reviews stream“. In: *Information Processing Management* 56 (cit. on p. 10).

Faridani, Siamak (2011). „Using Canonical Correlation Analysis for Generalized Sentiment Analysis, Product Recommendation and Search“. In: *Proceedings of the Fifth ACM Conference on Recommender Systems*. RecSys ’11. Chicago, Illinois, USA: ACM, pp. 355–358 (cit. on p. 22).

Faruqui, Manaal, Jesse Dodge, Sujay Kumar Jauhar, et al. (2014). „Retrofitting Word Vectors to Semantic Lexicons“. In: *CoRR* abs/1411.4166. arXiv: `1411.4166` (cit. on p. 22).

Glavas, Goran and Ivan Vulic (2018). „Explicit Retrofitting of Distributional Word Vectors“. In: *ACL* (cit. on p. 22).

Gurusamy, Vairaprakash and Subbu Kannan (Oct. 2014). „Preprocessing Techniques for Text Mining“. In: (cit. on pp. 41, 44, 45).

Hai, Zhen, Kuiyu Chang, and Jung-Jae Kim (Feb. 2011). „Implicit Feature Identification via Co-occurrence Association Rule Mining“. In: pp. 393–404 (cit. on p. 9).

Halder, Shayoni (Oct. 2019). „Textual Similarity Embeddings for Cross-Lingual Aspect Extraction“. MA thesis. Technical University of Munich (cit. on p. 82).

Hao, Shudong and Michael J. Paul (2018). „Learning Multilingual Topics from Incomparable Corpus“. In: *CoRR* abs/1806.04270. arXiv: 1806.04270 (cit. on p. 11).

Hartmann, Mareike, Yova Kementchedjhieva, and Anders Søgaard (2018). „Why is unsupervised alignment of English embeddings from different algorithms so hard?“ In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*. Brussels, Belgium: Association for Computational Linguistics, pp. 582–586 (cit. on p. 73).

he, Ruidan, Wee Sun Lee, Hwee Tou Ng, and Daniel Dahlmeier (Jan. 2017). „An Unsupervised Neural Attention Model for Aspect Extraction“. In: pp. 388–397 (cit. on pp. ix, 4, 5, 10, 31, 32, 45–47, 50, 51).

Hotelling, Harold (1936). „Relations Between Two Sets of Variates“. In: *Biometrika* 28.3/4, pp. 321–377 (cit. on p. 21).

Hu, Dichao (2018). „An Introductory Survey on Attention Mechanisms in NLP Problems“. In: *CoRR* abs/1811.05544. arXiv: 1811.05544 (cit. on p. 10).

Hu, Minqing and Bing Liu (2004). „Mining and Summarizing Customer Reviews“. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. KDD '04. Seattle, WA, USA: ACM, pp. 168–177 (cit. on pp. 4, 9).

Huber, Johannes and Myra Spiliopoulou (Jan. 2019). „Learning multilingual topics through aspect extraction from monolingual texts“. In: *Proceedings of the Fifth International Workshop on Computational Linguistics for Uralic Languages*. Tartu, Estonia: Association for Computational Linguistics, pp. 154–183 (cit. on pp. 10–12).

Iyyer, Mohit, Anupam Guha, Snigdha Chaturvedi, Jordan Boyd-Graber, and Hal Daumé III (June 2016). „Feuding Families and Former Friends: Unsupervised Learning for Dynamic Fictional Relationships“. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. San Diego, California: Association for Computational Linguistics, pp. 1534–1544 (cit. on p. 35).

Jagarlamudi, Jagadeesh and Hal Daumé (2010). „Extracting Multilingual Topics from Unaligned Comparable Corpora“. In: *Proceedings of the 32Nd European Conference on Advances in Information Retrieval*. ECIR'2010. Milton Keynes, UK: Springer-Verlag, pp. 444–456 (cit. on p. 11).

Jain, Rajat (May 2018). „Relation Extraction and Classification using Machine Learning“. MA thesis. Technical University of Munich (cit. on p. 39).

Jakob, Niklas and Iryna Gurevych (2010). „Extracting Opinion Targets in a Single- and Cross-domain Setting with Conditional Random Fields“. In: *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*. EMNLP '10. Cambridge, Massachusetts: Association for Computational Linguistics, pp. 1035–1045 (cit. on p. 10).

Jebbara, Soufian and Philipp Cimiano (2019). „Zero-Shot Cross-Lingual Opinion Target Extraction“. In: *CoRR* abs/1904.09122. arXiv: 1904.09122 (cit. on p. 12).

Joulin, Armand, Piotr Bojanowski, Tomas Mikolov, and Edouard Grave (2018). „Improving Supervised Bilingual Mapping of Word Embeddings“. In: *CoRR* abs/1804.07745. arXiv: 1804.07745 (cit. on p. 12).

Kasischke, Florian (Mar. 2019). „Descriptive Analytics for a NLP-Based Opinion Mining“. MA thesis. Technical University of Munich (cit. on p. 39).

Kessler, Jason and Nicolas Nicolov (Jan. 2009). „Targeting Sentiment Expressions through Supervised Ranking of Linguistic Configurations“. In: (cit. on p. 9).

Kumar, K.saravana and R.manicka Chezian (2012). „Article: A Survey on Association Rule Mining using Apriori Algorithm“. In: *International Journal of Computer Applications* 45.5. Full text available, pp. 47–50 (cit. on p. 4).

Lau, Jey, David Newman, and Timothy Baldwin (Jan. 2014). „Machine Reading Tea Leaves: Automatically Evaluating Topic Coherence and Topic Model Quality“. In: *14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014*, pp. 530–539 (cit. on pp. 27, 28).

Lei, Yaguo (Dec. 2017). „Individual intelligent method-based fault diagnosis“. In: pp. 67–174 (cit. on p. 24).

Li, Xin, Lidong Bing, Piji Li, Wai Lam, and Zhimou Yang (2018). „Aspect Term Extraction with History Attention and Selective Transformation“. In: *CoRR* abs/1805.00760. arXiv: 1805.00760 (cit. on p. 9).

Lim, Kar Wai and Wray L. Buntine (2016). „Twitter Opinion Topic Model: Extracting Product Opinions from Tweets by Leveraging Hashtags and Sentiment Lexicon“. In: *CoRR* abs/1609.06578. arXiv: 1609.06578 (cit. on p. 5).

Liu, Bing (2012). *Sentiment Analysis and Opinion Mining*. Morgan & Claypool Publishers (cit. on pp. 3, 4).

Liu, Bing, Minqing Hu, and Junsheng Cheng (May 2005). „Opinion observer: Analyzing and comparing opinions on the Web“. In: (cit. on p. 9).

Luong, Minh-Thang, Hieu Pham, and Christopher D. Manning (2015). „Effective Approaches to Attention-based Neural Machine Translation“. In: *CoRR* abs/1508.04025. arXiv: 1508.04025 (cit. on p. 25).

Manning, Christopher D., Prabhakar Raghavan, and Hinrich Schütze (2008). *Introduction to Information Retrieval*. Cambridge University Press (cit. on p. 45).

Mathapati, Savitha, S H Manjula, and Venugopal K R (2017). „Sentiment Analysis and Opinion Mining from Social Media : A Review“. In: *Global Journal of Computer Science and Technology* (cit. on p. 3).

McCormick, C (Jan. 2017). *Word2Vec Tutorial Part 2 - Negative Sampling*. URL: http://mccormickml.com/2017/01/11/word2vec-tutorial-part-2-negative-sampling/ (cit. on p. 17).

Meloun, Milan and J. Militky (2011). *Statistical Data Analysis: A Practical Guide*. Woodhead Publishing, Limited (cit. on p. 21).

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (2013a). „Distributed Representations of Words and Phrases and their Compositionality“. In: *CoRR* abs/1310.4546. arXiv: 1310.4546 (cit. on p. 17).

Mikolov, Tomas, Kai Chen, Gregory S. Corrado, and Jeffrey Dean (2013b). „Efficient Estimation of Word Representations in Vector Space“. In: *CoRR* abs/1301.3781 (cit. on pp. ix, 3, 6, 12, 14–16, 32).

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, pp. 262–272 (cit. on pp. 7, 26, 27).

Mount, John (Nov. 2014). *Approximation by orthogonal transform*. URL: `http://winvector.github.io/xDrift/orthApprox.pdf` (cit. on p. 20).

Newman, David, Jey Han Lau, Karl Grieser, and Timothy Baldwin (June 2010). „Automatic Evaluation of Topic Coherence“. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Los Angeles, California: Association for Computational Linguistics, pp. 100–108 (cit. on pp. 26, 27).

Pang, Bo and Lillian Lee (Jan. 2008). „Opinion Mining and Sentiment Analysis“. In: *Found. Trends Inf. Retr.* 2.1-2, pp. 1–135 (cit. on p. 9).

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. ix, 6, 12, 17, 18, 46, 51).

Popescu, Ana-Maria and Oren Etzioni (2005). „Extracting Product Features and Opinions from Reviews“. In: *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*. HLT ’05. Vancouver, British Columbia, Canada: Association for Computational Linguistics, pp. 339–346 (cit. on p. 9).

Poria, Soujanya, Erik Cambria, and Alexander Gelbukh (June 2016). „Aspect Extraction for Opinion Miningwith a Deep Convolutional Neural Network“. In: *Knowledge-Based Systems* 108 (cit. on p. 10).

Prabhakaran, Selva (Oct. 2018). *Lemmatization Approaches with Examples in Python*. URL: `https://www.machinelearningplus.com/nlp/lemmatization-examples-python/` (cit. on p. 45).

Qiu, Guang, Bing Liu, Jiajun Bu, and Chun Chen (Mar. 2011). „Opinion Word Expansion and Target Extraction Through Double Propagation“. In: *Comput. Linguist.* 37.1, pp. 9–27 (cit. on p. 9).

Reisenbichler, Martin and Thomas Reutterer (2019). „Topic modeling in marketing: recent advances and research opportunities“. In: *Journal of Business Economics* 89.3, pp. 327–356 (cit. on p. 5).

Röder, Michael, Andreas Both, and Alexander Hinneburg (2015). „Exploring the Space of Topic Coherence Measures“. In: *Proceedings of the Eighth ACM International Conference on Web Search and Data Mining*. WSDM ’15. Shanghai, China: ACM, pp. 399–408 (cit. on p. 80).

Rong, Xin (2014). „word2vec Parameter Learning Explained“. In: *CoRR* abs/1411.2738. arXiv: `1411.2738` (cit. on pp. 15, 16).

Rosset, Saharon, Ji Zhu, and Trevor Hastie (2003). „Margin Maximizing Loss Functions“. In: *Proceedings of the 16th International Conference on Neural Information Processing Systems*. NIPS’03. Whistler, British Columbia, Canada: MIT Press, pp. 1237–1244 (cit. on p. 34).

Rush, Alexander M., Sumit Chopra, and Jason Weston (2015). „A Neural Attention Model for Abstractive Sentence Summarization“. In: *CoRR* abs/1509.00685. arXiv: `1509.00685` (cit. on p. 10).

Schönemann, Peter H. (1966). „A generalized solution of the orthogonal procrustes problem“. In: *Psychometrika* 31.1, pp. 1–10 (cit. on pp. 19, 20).

Schouten, K. and F. Frasincar (2016). „Survey on Aspect-Level Sentiment Analysis“. In: *IEEE Transactions on Knowledge and Data Engineering* 28.3, pp. 813–830 (cit. on pp. 4, 9, 10).

Schouten, K., O. van der Weijde, F. Frasincar, and R. Dekker (2018). „Supervised and Unsupervised Aspect Category Detection for Sentiment Analysis with Co-occurrence Data“. In: *IEEE Transactions on Cybernetics* 48.4, pp. 1263–1275 (cit. on pp. 10, 31).

Sharma, Sagar (Sept. 2017). *Activation Functions in Neural Networks*. URL: `https://towardsdatascience.com/activation-functions-neural-networks-1cbd9f8d91d6` (cit. on p. 23).

Socher, Richard, Andrej Karpathy, Quoc Le, Christoper Manning, and Andrew Ng (Dec. 2014). „Grounded Compositional Semantics for Finding and Describing Images with Sentences“. In: *Transactions of the Association for Computational Linguistics* 2, pp. 207–218 (cit. on p. 35).

Song, Hye-Jeong, Byeong-Hun Yoon, Young shin Youn, et al. (2018). „A method of inferring the relationship between Biomedical entities through correlation analysis on text“. In: *Biomedical engineering online* (cit. on p. 22).

Stevens, Keith, Philip Kegelmeyer, David Andrzejewski, and David Buttler (2012). „Exploring Topic Coherence over Many Models and Many Topics“. In: *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. EMNLP-CoNLL '12. Jeju Island, Korea: Association for Computational Linguistics, pp. 952–961 (cit. on p. 26).

Su, Qi, Xinying Xu, Honglei Guo, et al. (2008). „Hidden Sentiment Association in Chinese Web Opinion Mining“. In: *Proceedings of the 17th International Conference on World Wide Web*. WWW '08. Beijing, China: ACM, pp. 959–968 (cit. on p. 4).

Tamura, Akihiro and Eiichiro Sumita (2016). „Bilingual Segmented Topic Model“. In: *ACL* (cit. on p. 11).

Tschannen, Michael, Olivier Bachem, and Mario Lucic (2018). „Recent Advances in Autoencoder-Based Representation Learning“. In: *ArXiv* abs/1812.05069 (cit. on p. 22).

Wang, Bin, Angela Wang, Fenxiao Chen, Yuncheng Wang, and C.-C. Jay Kuo (2019). „Evaluating Word Embedding Models: Methods and Experimental Results“. In: *CoRR* abs/1901.09785. arXiv: `1901.09785` (cit. on p. 7).

Wang, Wenya, Sinno Jialin Pan, Daniel Dahlmeier, and Xiaokui Xiao (2017). „Coupled Multi-Layer Attentions for Co-Extraction of Aspect and Opinion Terms“. In: *AAAI* (cit. on p. 11).

Wang, Yanbo (Jan. 2008). „Various Approaches in Text Pre-processing“. In: (cit. on p. 41).

Weng, Lilian (June 2018). *Attention? Attention!* URL: `https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html` (cit. on p. 25).

Weston, Jason, Samy Bengio, and Nicolas Usunier (2011). „Wsabie: Scaling Up To Large Vocabulary Image Annotation“. In: *Proceedings of the International Joint Conference on Artificial Intelligence, IJCAI* (cit. on p. 35).

Widmer, Christian (Apr. 2018). „Topic Modeling for Opinion Mining“. MA thesis. Technical University of Munich (cit. on p. 39).

Wu, Chuhan, Fangzhao Wu, Sixing Wu, Zhigang Yuan, and Yongfeng Huang (Jan. 2018). „A Hybrid Unsupervised Method for Aspect Term and Opinion Target Extraction“. In: *Knowledge-Based Systems* (cit. on p. 10).

Xing, Chao, Dong Wang, Chao Liu, and Yiye Lin (2015). „Normalized Word Embedding and Orthogonal Transform for Bilingual Word Translation“. In: *HLT-NAACL* (cit. on p. 21).

Yamada, Ikuya, Akari Asai, Hiroyuki Shindo, Hideaki Takeda, and Yoshiyasu Takefuji (2018). „Wikipedia2Vec: An Optimized Tool for Learning Embeddings of Words and Entities from Wikipedia“. In: *arXiv preprint 1812.06280* (cit. on pp. 64, 68).

Yin, Zi and Yuanyuan Shen (2018). „On the Dimensionality of Word Embedding“. In: *Proceedings of the 32Nd International Conference on Neural Information Processing Systems*. NIPS'18. Montréal, Canada: Curran Associates Inc., pp. 895–906 (cit. on p. 14).

Young, T., D. Hazarika, S. Poria, and E. Cambria (2018). „Recent Trends in Deep Learning Based Natural Language Processing [Review Article]“. In: *IEEE Computational Intelligence Magazine* 13.3, pp. 55–75 (cit. on p. 3).

Yu, Mo and Mark Dredze (2014). „Improving Lexical Embeddings with Semantic Knowledge“. In: *ACL* (cit. on p. 22).

Zhang, Yu and Weixiang Zhu (2013). „Extracting Implicit Features in Online Customer Reviews for Opinion Mining“. In: *Proceedings of the 22Nd International Conference on World Wide Web*. WWW '13 Companion. Rio de Janeiro, Brazil: ACM, pp. 103–104 (cit. on p. 4).

Zhao, Bing (Jan. 2006). „BiTAM: Bilingual Topic AdMixture Models for Word Alignment.“ In: (cit. on p. 11).

– (Jan. 2007). „HM-BiTAM: Bilingual Topic Exploration, Word Alignment, and Translation.“ In: (cit. on p. 11).

Zhou, X., X. Wan, and J. Xiao (2015). „CLOpinionMiner: Opinion Target Extraction in a Cross-Language Scenario“. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.4, pp. 619–630 (cit. on p. 11).

Zhu, Yonghua, Xun Gao, Weilin Zhang, Shenkai Liu, and Yuanyuan Zhang (Nov. 2018). „A Bi-Directional LSTM-CNN Model with Attention for Aspect-Level Text Classification“. In: *Future Internet* 10, p. 116 (cit. on p. 11).

Zhu, Zede, Miao Li, Lei Chen, and Zhenxin Yang (Aug. 2013). „Building Comparable Corpora Based on Bilingual LDA Model“. In: vol. 2, pp. 278–282 (cit. on p. 11).

List of Figures

List of Tables