Universiteit Leiden

Master Computer Science

Aspect-Based Sentiment Analysis on Dutch Patient Experience Survey Data

Name:	Murad H. Bozik
Student ID:	s2619822
Date:	8/07/2022
Specialisation:	Artificial Intelligence
1st supervisor:	Suzan Verberne, Assoc. Prof.
2nd supervisor:	Ilse Kant, PhD
Daily supervisor:	Marieke van Buchem, PhD(c)

Master's Thesis in Computer Science

Leiden Institute of Advanced Computer Science (LIACS)
Leiden University
Niels Bohrweg 1
2333 CA Leiden
The Netherlands

Abstract

Aspect-based sentiment analysis (ABSA) extracts sentiments concerning a given aspect in a sentence. Models perform less well in the ABSA task on free-text in health care and well-being. Recent studies showed that higher performances are achievable by integrating syntactic dependencies into representation learning. This research investigates the benefits of syntactic relations on graph convolution (GCN) and relational graph attention (RGAT) networks for patient experience survey data collected from three hospitals. We compared our implementations against state-of-the-art and classical models. In addition, we explore the transferability of the method using cross-hospital analysis.

Contents

Chapter 1

Introduction

1.1 Problem Statement

Patients receive a personal service from hospitals. Understanding whether they are satisfied with the service they receive enhances the quality of this individual service. Hospitals create surveys to evaluate their service quality. Most questionnaires contain closed-ended questions and may not address the points the patients want to express. This is why the majority of Dutch hospitals send a questionnaire called Patient Experience Monitor (PEM) [5] to patients discharged from the hospital to fill out. PEM questionnaire uses two open-ended questions in addition to closed-ended questions: "What went well in the hospital?" and "What could be improved?". Free-text responses made by patients contain valuable information but are time-consuming to analyze. The answers to the first question comprise the positive topics, and the column of the second question mentions the topics with a negative sentiment.

People tend to answer open-ended questions in various ways. Some write positive and negative comments in the same section, and some state the opposite sentiment in a particular column. The inconsistency and obscurity emerge the problem of sentiment classification. Extracting the topics mentioned in the comments is another facet of the problem. We applied topic modeling to understand what the topics were. We manually annotated the identified topics of a small portion of the dataset. Then, we employed a custom Named Entity Recognition (NER) model trained on the manually annotated dataset for the extraction of the aspects from the whole dataset. Aspect-based sentiment analysis (ABSA) explores the polarity of mentioned aspects in a sentence. Even though the aspect-based sentiment classification task is a straightforward problem to solve, success in this task in the healthcare field lags behind other domains such as restaurant reviews and social media analysis [32].

1.2 The Goal of Research & Contributions

Researchers applied different techniques in search of better performances in the aspect-based sentiment analysis task. A recent approach is the integration of syntactic information into the model representation [27] [30] [19] [3] [31]. In this research, we implemented two approaches to explore the benefits of dependency information at the sentence level. We compared these approaches with classical methods (SVM, Logistic Regression, ...) and state-of-the-art BERT models. We seek to answer the following research questions:

- Does the integration of syntactic dependencies provide any leverage compared to other methods in the Dutch language?

- Is this approach suitable for providing actionable insight into improving health care and well-being by uncovering the strengths and weaknesses of the hospital's services?

In this study, we worked with the patient experience questionnaire data of 3 hospitals. The development dataset was created by combining the data of the first and second hospitals. The development dataset contains 55507 responses. At the character level, the responses have mean, median, and standard deviation of 106, 63, and 140, respectively. The data of the third hospital were kept separately for cross-hospital analysis. The cross-hospital dataset includes 10297 responses with a mean, median, and standard deviation of 92, 54, and 123, respectively.

We summarize our contributions as follows.

⋆ We compared the contributions of syntactic dependencies in the aspect-based sentiment classification task in the Dutch language.

⋆ We conducted a cross-domain analysis by applying the relational graph attention model (RGAT) in the healthcare domain.

⋆ We presented a cross-hospital analysis by extracting the positive and negative aspects from the hospitals' data and comparing hospitals.

1.3 Overview of the Thesis

The thesis is outlined as follows. Chapter 2 explains similar approaches and backgrounds to the methods implemented in the following sections. Chapter 3 describes the data sources and demonstrates the preprocessing steps. Chapter 4 clarifies the implemented approaches and experiment details. Chapter 5 presents models' performances and provides cross-hospital and error analysis. Chapter 6 shows the challenges and limitations encountered during the research. The future works and the utility of this research for the hospital are explained in this chapter. Finally, chapter 7 briefly summarizes the thesis and the contributions.

Chapter 2

Background and Related Work

Sentiment analysis evaluates the polarity of the sentences. The opinions belonging to different polarity classes can occur in the same sentence. Aspect-based sentiment analysis (ABSA) provides a better sentiment analysis by paying attention to the respective polarities of each aspect in a sentence.

In the medical domain, sentiment analysis can provide doctors and hospital managers insights on how to improve healthcare service. Cammel *et al.* [7] investigates the benefits of NLP techniques to extract valuable insights from the PEM questionnaire to improve healthcare service. They divide free-text responses into topics and subtopics using non-negative matrix factorization (NMF) and extract sentiments from free-text using word frequency tables. Nemes *et al.* [23] attempt sentiment analysis about COVID-19 on Twitter data. They first apply a sentiment analysis, then information extraction, and named entity recognition to provide a deeper analysis of the subject. They compare BERT, RNN, nltk, and TextBlob sentiment analyses and show that BERT performs best among them.

Žunić *et al.* [32] perform a systematic review of sentiment analysis in the healthcare domain and state that the performances lag behind other domains. Gräßer *et al.* [14] perform an aspect-based sentiment analysis on the healthcare domain and address the challenges of insufficiency of annotated data. They perform sentiment analysis using Logistic Regression on n-gram representations of free-text responses. They discuss transfer learning from similar domains to improve performance in the healthcare domain.

In order to enhance the aspect-based sentiment classification performance, researchers experimented with both data-centric and model-centric approaches. Ruder *et al.* [24] propose a hierarchical bidirectional LSTM model for a restaurant review dataset. Wang *et al.* [28] integrate an attention mechanism into the LSTM model, which focuses on particular parts in a sentence to provide more granular sentiment analysis. Bao *et al.* [4] combine attention-based LSTM with lexicon features extracted from various sources.

Sun *et al.* [26] attempt to convert the ABSA task into question answering (QA) and natural language inference (NLI) tasks by introducing an auxiliary sentence. They incorporate the BERT model in their approach. Xu *et al.* [29] introduce a new task called review reading comprehension (RRC), and they approach ABSA as a version of the RRC task. They show that post-training on fine-tuned BERT model improves performance. Hoang *et al.* [18] use the BERT model as a contextual encoder and improve its representation with additional text as in the paper by Xu *et al.* [29].

De Clercq *et al.* [8] presented an ABSA pipeline for Dutch restaurant and smartphone reviews. They perceived ABSA as three subtasks: aspect term extraction, aspect category classification, and aspect polarity classification. Their pipeline utilized an SVM classifier for the third subtask and achieved 81.23 accuracy. In their second paper [9], they evaluated their pipeline in the banking, retail, and human resources domains and achieved 86.8, 88.9, and 86.7 success, respectively.

Recently, integrating syntactic information got attention in enhancing representation learning. Sun *et al.* [27] propose a graph convolution network (GCN) combined with BiLSTM. BiLSTM learns the dependency tree representation, and GCN performs convolutions over the nodes of BiLSTM output. They apply average pooling operation to aggregate final embeddings into a dense vector for the classification layer. Similarly, Zhang *et al.* [30] used BiLSTM and GCN in combination. They used BiLSTM to capture contextual information from embeddings. They feed GCN with BiLSTM hidden layer, and after applying convolution over the dependency graph, they select aspect-specific features with a masking layer. Aspect-specific information is fed to LSTM back to enhance aspect-specific contextual information and produce the final representation for polarity classification.

The dependency information can be represented as a graph. Thus, the researchers implement neural networks to operate on graph representations. Huang *et al.* [19] present a syntax-aware graph attention network (TD-GAT). Graph attention network (GAT) uses a dependency tree as a graph and updates the nodes with multi-head attention weights. They experiment with combining GAT with LSTM and BERT models and compare them with LSTM, CNN, and SVM models.

Žunić *et al.* [31] implement a GraphSAGE-GCN model. GraphSAGE-GCN model is developed by Hamilton *et al.* [16] to learn an inductive representation from graphs. Hamilton *et al.* propose various aggregator functions to propagate information from local neighborhoods of the nodes. Žunić *et al.* take their approach as their base model, apply a summation aggregator, and evaluate it in the medical domain. Žunić's paper uses the drug review dataset and makes automatic annotations for the aspects. All the aspects consist of only one word. So, their approach does not consider multi-word aspects. In our research, we implement the model from Žunić *et al.* paper [31]. We change the model by implementing average pooling over aspect tokens. This modification makes our model more flexible and suitable for consecutive and non-consecutive multi-word aspects.

Another approach we evaluate in this research regarding syntactic dependency integration is the relational graph attention model (RGAT). Bai *et al.* [3] propose a relation aware graph attention framework. The framework computes the final representation from contextual and syntactic representations. BERT or a BiLSTM model is used as a contextual encoder, and syntactic representation is calculated through a transformer model. The transformer model consists of a multi-head attention layer, a point-wise fully connected layer, and normalization layers with skip-gram connections. In our evaluation, we replace the contextual encoder with fine-tuned BERTje model.

Chapter 3

Data collection & construction

3.1 Data Source & Construction of the Initial Dataset

Data consists of questionnaires from three hospitals. Due to privacy concerns, we will refer to these hospitals as Hospital 1 (H1), Hospital 2 (H2), and Hospital 3 (H3). The hospitals use three types of forms for collecting patient experience. These are PEM, BeterMeter, and H1 List.

- PEM questionnaire is the recent questionnaire used by most Dutch hospitals.
- BeterMeter is a department-specific questionnaire created by Hospital 1.
- H1 List is the older version of PEM developed by Hospital 1.

The data from H2 and H3 hospitals are PEM questionnaires. H1 data includes all three types of forms.

These forms have two open-ended questions: What went well? What could be better? Answers to the questions are free-text responses. In order to combine different forms, some columns are treated as positive labeled responses, others as negative. Table 3.1 shows how the columns are labeled. The data from H3 was held out for cross-hospital analysis. The data from H1 and H2 hospitals are combined. All the data is pre-anonymized by the hospitals.

Forms	***Positive***	***Negative***
PEM H1 (2020)	Wat gaat goed op poli	Wat kan beter op poli
	Wat gaat goed op verpleegafdeling	Wat kan beter op verpleegafdeling
	Wat gaat goed op poli – ouders	Wat kan beter op poli – ouders
2019 H1	Wat wilt u nog vertellen	Wat kan beter
H1 List	Wat ging goed?	Wat kon beter?
PEM H2	Wat gaat goed	Wat kan beter
PEM H3	Wat gaat goed	Wat kan beter

Table 3.1: *How the columns of forms are treated to create a combined dataset*

3.2 Preprocessing Steps

Data includes not only Dutch but also English responses. We identified 480 English responses using Google Translation API and removed them from the data before preprocessing. We followed two preprocessing steps, one for topic modeling and the other for sentiment analysis tasks. Since the topic modeling task is a statistical analysis method, it needs a much plain version of the text. Topic modeling is used to get an intuition for aspects and select keywords for the aspects. For the sentiment analysis task, it is more beneficial if the data is closest to the original form. Therefore, some preprocessing steps remain the same, while others differ. We named the diverging steps as normalization steps. Overall, preprocessing consists of 17 steps. The preprocessing pipeline is constructed as a function that takes a sentence as input, applies consecutive steps, and outputs the processed/normalized version of the text. Below we present the consecutive steps and some examples to clarify them. Normalization steps include 6th, 11th, 13th, 15th, and 16th steps. These steps are discarded for the sentiment analysis task.

1. Removing spaces from the beginning and the end of a sentence
2. Fixing errors caused by HTML parsing
   - &#128077;&#128077; ⇒ 👍 👍
   - Ik vond &#8216;t te snel dat ⇒ Ik vond ‘t te snel dat
3. Unifying different types of apostrophes
4. Removing newline characters in the text
5. Replacing URLs with keywords as [url]
6. Removing HTML-like tags from the text
7. Changing possible signs/words using regex
   - +_ ⇒ plusminus
   - % ⇒ percent
8. Expanding the contractions
   - m'n gemak ⇒ mijn gemak
   - zo'n goede plek ⇒ zo een goede plek
9. Expanding the abbreviations
   - De tijd op de OK ⇒ De tijd op de operatie kamer
   - nvt. ⇒ niet van toepassing
10. Splitting slashes to tokenize the text later correctly
11. Removing numbers from the text
12. Converting emojis into text

   - Toppie 👍 ⇒ Toppie :thumbs_up:
   - respect 😎 ⇒ respect :smiling_face_with_sunglasses:

13. Removing punctuations
14. Tokenization using nltk library [1]
15. Spell checking using Hunspell library
16. Lemmatization using nltk library [2]
17. Filtering tokens by length (We kept the tokens with at least two characters)

In order to preserve consistency with tokenization in the subsequent process, we unified the apostrophe styles (step 3), expanded the contractions in the text (step 8), and added whitespace before and after the slashes (step 10) in the text. For the expanding abbreviations we created a domain-specific dictionary using wikipedia[3] and an open-source website[4]. Finally, after 16 steps, we hold tokens of at least two characters. The final list of tokens are combined to create processed/normalized version of the input sentence.

We needed dependency tree representations for the approaches: Relational Graph Attention Model (RGAT) and Graph Convolutional Network (GCN). We used the dependency parsing pipeline from the stanza library as a subsequent process. This pipeline incorporates its own tokenization, pos tagging, lemmatization, and dependency parsing steps. Stanza package utilizes the Universal Dependencies (version 2) structure.[5]

[1] https://www.nltk.org/api/nltk.tokenize.html

[2] https://www.nltk.org/_modules/nltk/stem/wordnet.html

[3] https://nl.wikipedia.org/wiki/Lijst_van_afkortingen_in_het_Nederlands

[4] https://verpleging-verzorging.nl/afkortingen-zorgsector/

[5] The detailed explanation of the Universal Dependency structure can be found on https://universaldependencies.org

Chapter 4

Methods

In this chapter, we introduce the methods used during the research. Section 4.1 explains the intuition behind the aspects, exploration through topic modeling, and annotation of the selected aspects. Section 4.2 presents the automatic annotation process via custom NER model predictions. Section 4.3 describes the approaches used in the sentiment analysis task.

4.1 Aspect Annotation

4.1.1 Intuition for Aspect Annotation

The normalized dataset has primarily short texts. Figure 4.1 shows the distribution of binned word counts. In order to identify the aspects in the normalized dataset, we intended to apply topic modeling and extract the group of words for the topics. In the case of non-overlapping topics, the collection of words for each topic would be used as a seed for selecting responses with intended aspects. Later the annotation would be performed on selected responses.

4.1.2 Specific Preprocessing (Manually removing meaningless responses)

Before applying Topic Modeling, we looked at the most frequent short comments to understand what to expect from the topics. Tables 8.5-8.6 demonstrates the top-10 comments for negative and positive columns. By looking at the most frequent short comments, we realized there are many meaningless comments in terms of sentiment. The positive column was relatively less noisy than the negative one. We created a list of unique comments up to 5 words in length for each column. We manually read and identified the meaningless comments in those lists. For example, the most frequent examples for the question ”What went wrong?” were [niets, geen opmerkingen, alles ging goed, zou het niet weten, ik zou het niet weten]. Comments like these examples are considered meaningless for the respective columns. Among the comments up to 5 words, we found a total of 1588 unique comments meaningless and removed 6225 comments from the data. After that, we applied topic modeling. Tables 8.7-8.8 presents the top-10 comments up to 3 words in length after removing the manually identified meaningless comments. These most frequent comments give an idea about what the aspects should be. In order to maximize the benefit of topic modeling, we took additional preprocessing steps and normalized the data to its simplest form. Following the additional preprocessing steps explained in section 3.2, we were able to obtain more expressive word frequencies for the topics.

4.1.3 Topic modeling & Identifying the Aspects

After cleaning the comments, we applied two techniques: Non-Negative Matrix Factorization (NMF) [20] and Latent Dirichlet Allocation (LDA) [6]. Both techniques were performed on a small portion of the dataset and the whole dataset. The small portion of the dataset is less noisy compared to the whole dataset as we manually clean up comments up to 5 words. We tried two implementations: standard LDA model from Gensim package[1] and LDA Mallet [21]. The LDA Mallet implementation achieved a better coherence score, and the topic visualization was more distinct. Figure 4.2 illustrates the word clouds for the negative and positive columns of the small portion. Appendix A includes the tables and graphs regarding topic modeling analysis. We expected to use topic words to define the aspects. Then, we planned to use the most common words for each aspect as seeds to select comments to be annotated with the aspects. However, extracting aspects using topic modeling did not produce the desired outcome due to overlapping topics. The most common words were repeated on multiple topics. Nonetheless, it was helpful to understand what aspects should be. We chose the aspects by investigating the most frequent words that indicate different topics and combining the information with our prior data exploration.

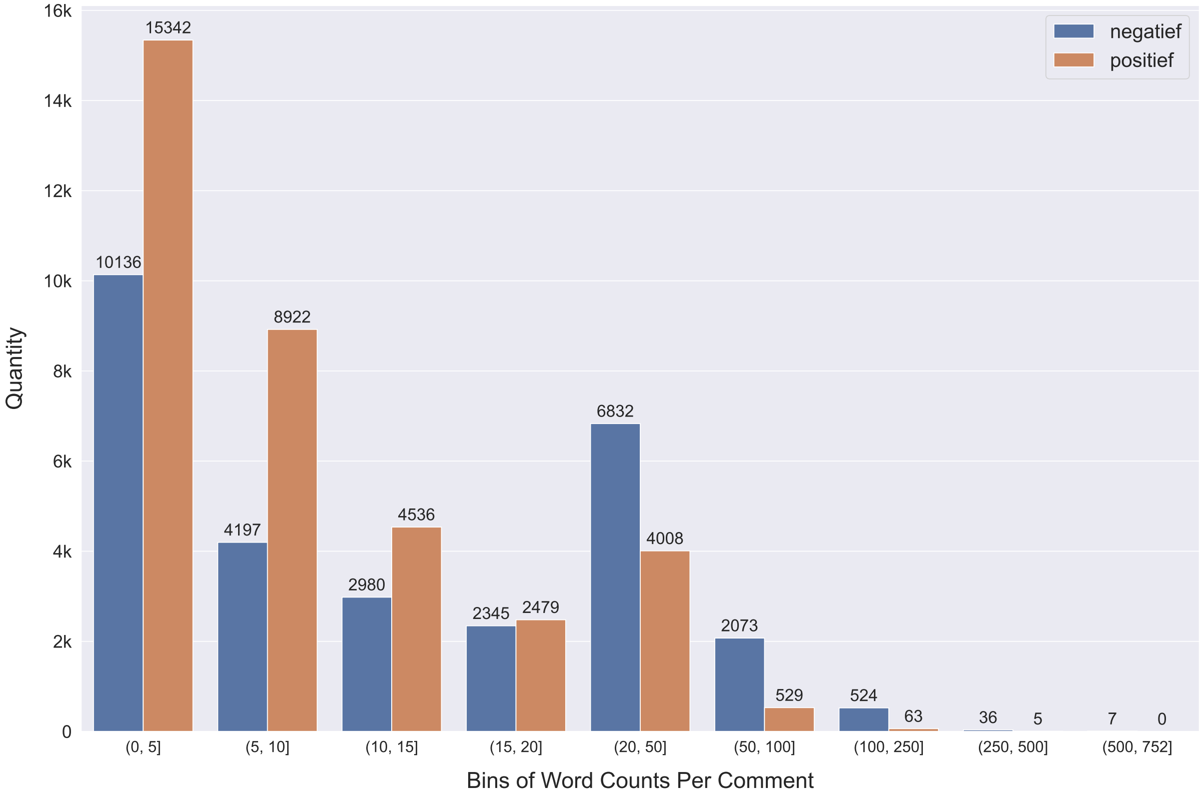
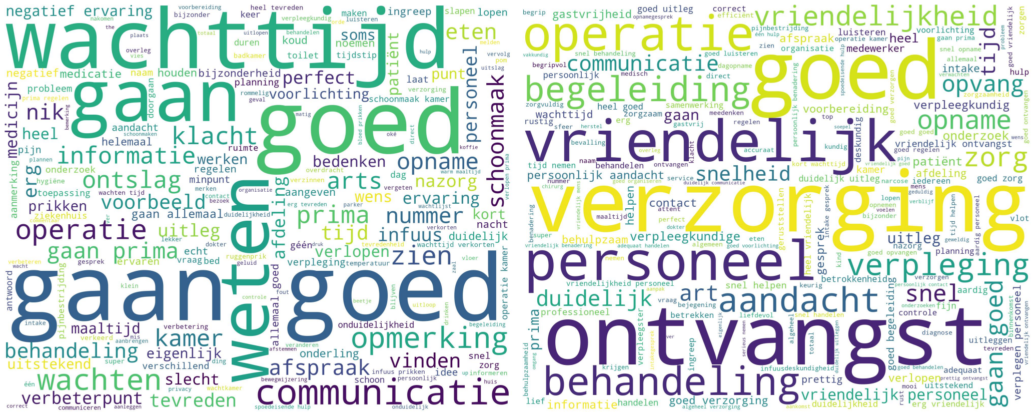


Figure 4.1: *Word Count Distribution*

Based on the observation, we selected the following aspects: Wachttijd (waiting time), Communicatie (communication), Eten (food), Behandeling (treatment), and Schoonmaken

[1] https://radimrehurek.com/gensim/



(a) *Negative Words*

(b) *Positive Words*

Figure 4.2: *Word Clouds of Negative and Positive Columns*

(cleaning) as negative aspects; Vriendelijkheid (kindness), Communicatie (communication), Snelheid (speed), Zorg (care), and Personeel (staff) as positive aspects.

4.1.4 Filtering Data with Keywords & Making Aspect Annotations

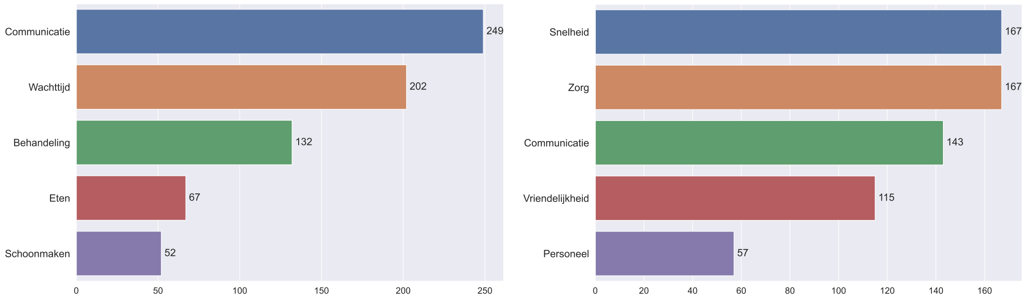
We aim to create an annotated dataset for aspect extraction while keeping the sentence length and sentiment label statistics the same as the entire dataset. We first defined a few words to describe each aspect. We then created a candidate pool by collecting responses containing these words using regular expression patterns. From this pool, we selected responses with only one sentence to be annotated, keeping the word length and label distributions the same as the whole data set. Tables 8.19 and 8.27 show the strings used to generate candidate pools.

We utilized the doccano[2] annotation tool. Appendixes E and F explain the guidelines we followed while annotating aspects. Unlike in the approach of Žunić *et al.* [31], our aspects are composed of a varying number of words. Sometimes the aspects include one word, sometimes up to 6 words. For instance, in the response "Had to be present very early, so had to wait a long time for surgery," we annotated "had to wait a long time" as our "waiting time" aspect. In order to simplify the annotation process, we assumed that each sentence contained only one aspect. Table 4.1 shows the distribution of the manually annotated dataset. Figure 4.3 presents the distribution of aspects over positive and negative responses.

	Positive	***Negative***	***Total***
Train	414	450	864
Validation	104	112	216
Test	131	140	271
Total	649	702	1351

Table 4.1: *Dataset Distribution*

[2] https://github.com/doccano/doccano



(a) *Negative Aspects*

(b) *Positive Aspects*

Figure 4.3: *Distribution of Aspects in Manually Annotated Dataset*

4.2 Named Entity Recognition For Extracting Aspects

4.2.1 Creating Dataset for Named Entity Recognition

We manually annotated only a small portion of the dataset. Then, in order to increase the annotation, we trained a named entity recognition (NER) model using flair NLP library to automatically extract all aspects from the large dataset.[3] First, we prepared a dataset in CoNLL-2003 [25] format. The annotation results of doccano is a list of tuples : [(start_index, end_index, Aspect_name), ...]. The aspect indexes refer to the character index in the corresponding sentence. We first converted these indexes into token ids, and then we created B- and I-tags for each aspect name using the token ids. In order to eliminate tokenization differences for later processes, we used the Stanza library to tokenize the sentence and the nltk library for extracting POS tags. Unlike the official CoNLL-2003 format, we specify the dataset with three columns as explained in the flair sequence labeling dataset[4]. Each line includes token, POS tag and BIO-annotated NER tag. They are separated with a space, and there is a blank line between each sentence. This process was repeated for each split of the small dataset.

4.2.2 Training Named Entity Recognition Models

The open-source Flair library provides a unified interface for different embeddings [1]. It is possible to use different embeddings in combination as stacked embeddings. This makes it easy to experiment with various types.

Flair creates a sequence tagger model for the NER task. Sequence tagger models consist of an embedding layer, two dropout layers (word_dropout and locked_dropout), an embedding projection layer (embedding2nn), Bidirectional LSTM, and a linear layer followed by Conditional Random Fields (CRF). Since CRF is used in this downstream task, standard cross-entropy loss is not suitable. The sequence tagger uses ViterbiLoss for loss calculation. ViterbiLoss is the sum of the negative log-likelihoods of all tag sequences minus the negative log-likelihood of the true tag sequence called - the gold score[5]. The only task for the practitioners is to

[3] https://github.com/flairNLP/flair

[4] https://github.com/flairNLP/flair/blob/master/resources/docs/TUTORIAL_6_CORPUS.md#reading-your-own-sequence-labeling-dataset

[5] Simpler explanation can be found on https://github.com/sgrvinod/a-PyTorch-Tutorial-to-Sequence-Labeling#viterbi-loss

change the embedding layer using the unified interface of the Flair library.

We experimented with two models. In the first model, we stacked Fast-text embeddings [15], forward and backward Flair embeddings [2]. Flair embeddings are encoder/decoder-style language models. In the second model, we insert a HuggingFace BERT model ("bert-base-dutch-cased") [10] into Flair pipeline as an embedding layer.

4.2.3 Automatically Annotating Complete Dataset

We automatically annotated the whole dataset using our custom NER model. Since the NER model does not depend on the label of the column (positive/negative), it can predict the aspects of both negative and positive responses. So, the previously selected negative aspects are not necessarily seen only in negative responses. We selected aspects based on their frequencies in negative and positive responses. Figures 4.4a and 4.4b demonstrate the aspect distribution percentages of the H1&2 dataset and H3 dataset, respectively.

The annotation of the whole dataset reflects the same bias as the manually annotated data. Consistency shows that the aspects we select are representative of the whole dataset. The aspects selected for negative responses are also more common in negative responses across the entire dataset. Positive aspects are also seen more frequently in positive responses.

We also annotated the H3 dataset for cross-hospital analysis purposes. The consistency can also be seen in this dataset. The only difference was in the communication aspect. Hospital 3 seems better in communication aspects as opposed to Hospital 1&2.

4.3 Baselines for Sentiment Analysis

4.3.1 Frequency Based Approach

We followed a frequency-based approach to create a simple baseline. In this approach, we create a weight vector θ, and we fit this weight vector to the training set for a certain amount of iterations. The updated weight vector is used to predict the sentiment for the new data. The below equation shows the decision function for this approach.

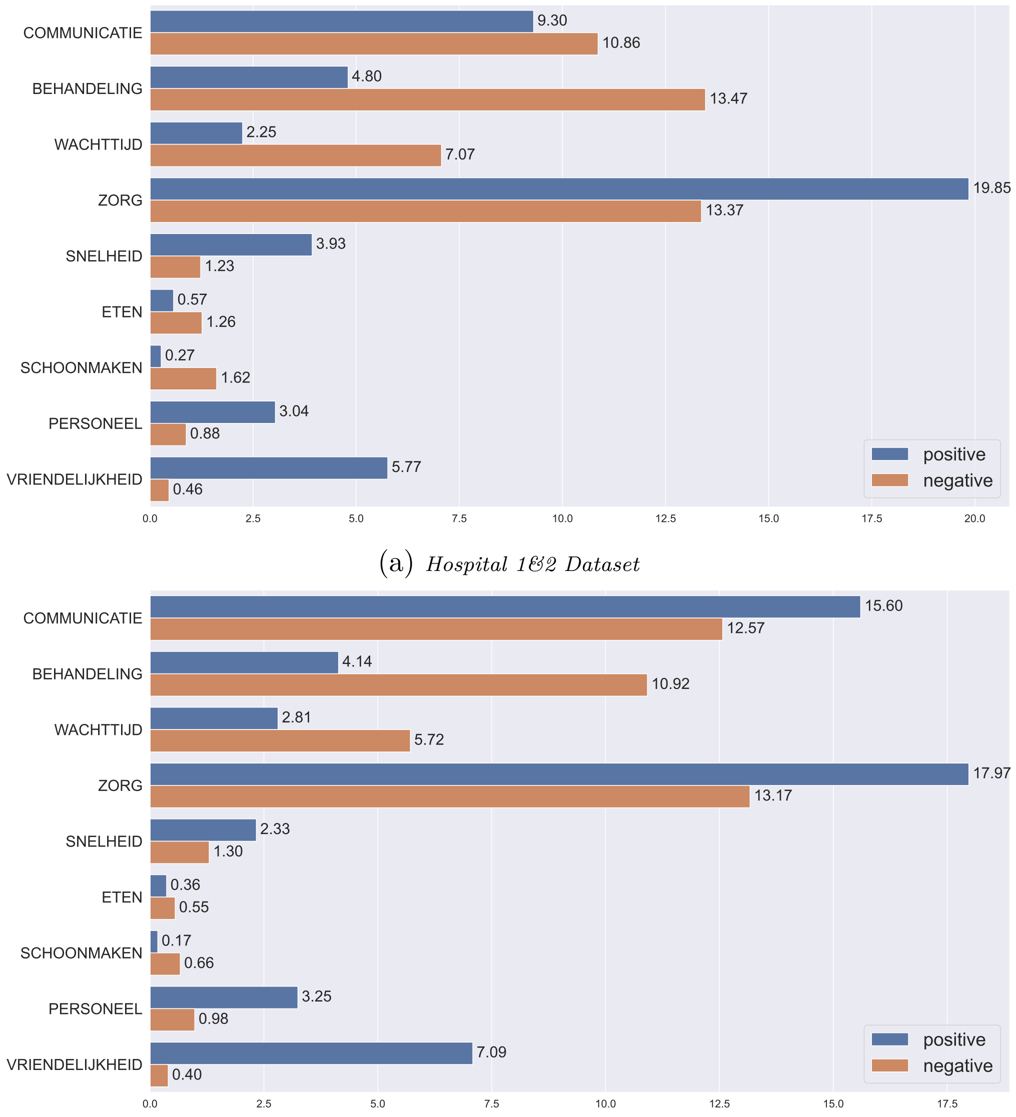
$$\hat{y} = \sigma(x \cdot \theta)$$

First, we applied a simple preprocessing step that includes removing punctuation and stop words, then taking the stem of the words using PorterStemmer from the nltk library. After preprocessing, we created a dictionary that maps (word, sentiment) tuple to its frequency.

Each sentence is represented by a vector (x): [Bias Term, Positive Frequency, Negative Frequency]. Feature vector has (m, 3) dimension shape. We defined a weight vector (θ) with a (3, 1) shape. Then we applied gradient descent to update θ. We applied 1500 iterations. In each iteration, we calculated z: the dot product of input and weight, then fed sigmoid function: $h : \sigma(z)$. θ is updated with following formula where α refers to learning rate, m indicates the number of training samples and y refers to target vector;

$$\theta := \theta - \frac{\alpha}{m} \times (x^T \cdot (h - y))$$

For the calculation of cost, we used cross-entropy loss.



(a) *Hospital 1&2 Dataset*

(b) *Hospital 3 Dataset*

Figure 4.4: *Aspect Distribution Across Datasets*

4.3.2 Conventional Methods

We compared neural-network model performances with following methods:

- Linear SVM
- Naive Bayes
- Logistic Regression
- Decision Tree
- Random Forest
- Ada Boost

For each method, we represented input text using TF-IDF features. [6] We used default parameters for all classical models except the following changes. For SVM, we set the class_weight parameter to balanced, and for logistic regression, we increased the iter_max parameter to 1000 for the automatically annotated dataset.

4.3.3 BERT-Based Models

For the state-of-the-art model comparison, we experimented with three BERT-based models:

- BERTje: Dutch BERT Model [10]
- mBERT: Multilingual BERT-uncased Model [12]
- RobBERT_v2: Dutch RoBERTa-based Language Model [11]

All BERT-based models are pre-trained models. We fine-tuned them using our datasets for the sentiment classification task. These models operate on the input text sequence as a whole and do not utilize aspect annotation or explicit information such as dependency relations and positional embeddings in the polarity classification. We first trained and evaluated models on the manually annotated dataset. We saw the performance of RobBERT was the worst among them. Due to computational cost, we selected mBERT and BERTje to train/test on the automatically annotated dataset.

4.3.4 Graph-SAGE GCN

Graph convolution networks (GCN) are suitable for utilizing dependency graphs as input. GCN model outputs the updated version of the input representations. It essentially creates a function for the input graph.

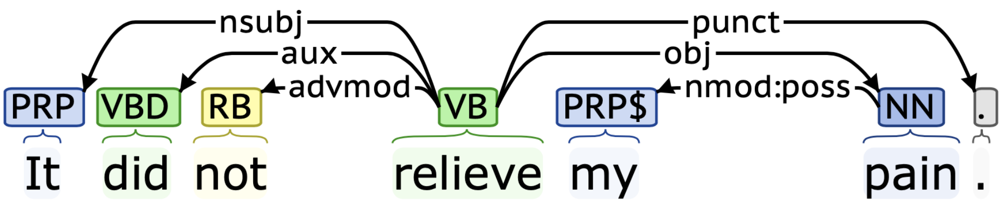


Figure 4.5: *Dependency Tree Example from the paper [31]*

We needed to parse dependency relations to feed the Graph-SAGE GCN (GS-GCN) model. We used Stanza library to create an undirected dependency graph representation of normalized text responses. Stanza represents the syntactic dependency relations in the Universal Dependencies (UD) formalism[7]. Figure 4.5 illustrates the example of a dependency tree.

Parsed dependency relations have the following attributes: id, word, head_id, head, deprel. Head id is 0 if the word itself is the root of the dependency tree. Head refers to the governor word. Deprel refers to dependency relations. The relations indicate directions such as word to

[6] https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfTransformer.html

[7] https://universaldependencies.org/

head. We utilized word_id and head_id attributes to create tuples of dependency relations. We flipped the dependency relations and concatenated them with the original order. For example; a dependency relation (a→b) became [(a→b), (b→a)]. The purpose of concatenation of inverted order is to convert the directed graph into an undirected one. Because the aggregator function is one-way operation. It takes target (b) and make the aggregation using the source (a) as index. In order to apply the aggregation to an undirected graph, we need to provide the reverse order along with the original order. The resulting graph is undirected. The words are represented as vertices, and created tuples are represented as edges.

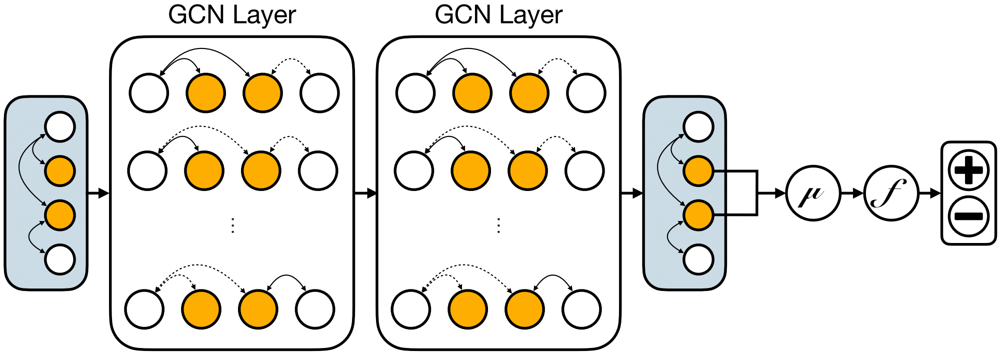


Figure 4.6: *GS-GCN Model Diagram*

Figure 4.6 displays the GS-GCN architecture. The input to GS-GCN is 300-dimensional Fasttext [22] embeddings of tokens. Embeddings are updated during GCN layers using the following function to aggregate information from neighbor tokens (heads).

$$h_i^t = \sigma\left(W^t \cdot \text{concat}\left(h_i^{t-1}, \text{aggregate}\left(h_j^{t-1}, \forall j \in N(i)\right) + b^t\right) \tag{4.1}$$

In the equation 4.1 *i* represents a particular vertex in the graph, while $N(i)$ is a set of neighbours of vertex i and *t* refers to layer number. The aggregate function first repeats/selects the embeddings based on vertices and then sums its neighbors. The aggregate function is handled using the scatter_add function[8] from the torch_scatter library. The hidden state h_i^t is computed by concatenating the aggregation result with the previous hidden state horizontally and then applying a non-linear RELU activation function to this output.

The equation 4.1 shows one-step information aggregation. The model applies the aggregation step twice, meaning that the GS-GCN model utilizes the information from second-order neighbors. After GCN Layers, a dropout with a 0.2 rate has been applied. Unlike Zunic *et al.*, our aspect annotations consist of multiple words. For this reason, we selected the vertices which belong to our aspect annotations, then took their average and fed the final linear layer to decrease output to a 2-dimensional vector. The softmax layer performed the final classification among negative and positive labels.

4.3.5 Relational Graph Attention Network (RGAT)

Bai *et al.* [3] proposed the Relational Graph ATtention (RGAT) method, which uses the dependency label information in addition to the dependency tree. They investigated the benefits

[8] https://pytorch-scatter.readthedocs.io/en/latest/functions/scatter.html

of dependency label integration with various architectures. We selected the model architecture (RGAT-BERT) with the best performance and adjusted it to our framework.

The representation of their data is in JSON format. All text includes only one sentence, and the target representation in this approach aligns with our aspect annotations. So, we converted our dataset into their format.

The model's input is a tuple of target, sentence, and dependency tree of the sentence. The tuple is denoted as as a triplet: $\langle \mathcal{T}, \mathcal{S}, \mathcal{G} \rangle$, where $\mathcal{T} = \{w_i, w_{i+1}, \ldots, w_{i+m-1}\}$ refers to the target word sequence, m denotes to the length of target mention, $\mathcal{S} = \{w_1, w_2, \ldots w_n\}$ represents a sentence, where n is the length of the sentence.

The dependency graph $\mathcal{G}$ is defined as a tuple $(\mathcal{V}, \mathcal{A}, \mathcal{R})$ where $\mathcal{V}$ is a list of vertices with length n. $\mathcal{A}$ denotes to the edges as adjacency matrix $\mathcal{A} \in \mathbb{R}^{n \times n}$. $\mathcal{A}_{ij} = 1$ if there is a dependency relation between word w_i and w_j, and $\mathcal{A}_{ij} = 0$ otherwise. $\mathcal{R}$ is a label matrix, where $\mathcal{R}_{ij}$ equals to the corresponding label of the relation in $\mathcal{A}_{ij}$, if $\mathcal{A}_{ij} = 1$, and if there is no relation then $\mathcal{R}_{ij} = None$.

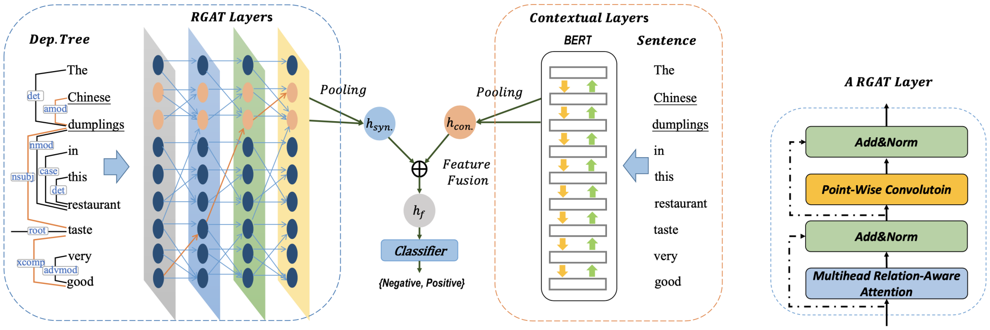


Figure 4.7: *RGAT Model Diagram from original paper [3]*

The model consists of three parts: the Contextual Encoder, Relational Graph Attention (RGAT) Encoder, and a classifier. Figure 4.7 shows the overall framework for this approach. Basically, the RGAT Encoder encodes the dependency tree along with the label information. The contextual encoder, which is a BERT model, encodes the sentence itself. Before feeding the classifier, the model combines the encodings using a feature fusion operation, and the classifier predicts the sentiment polarity of the sentence.

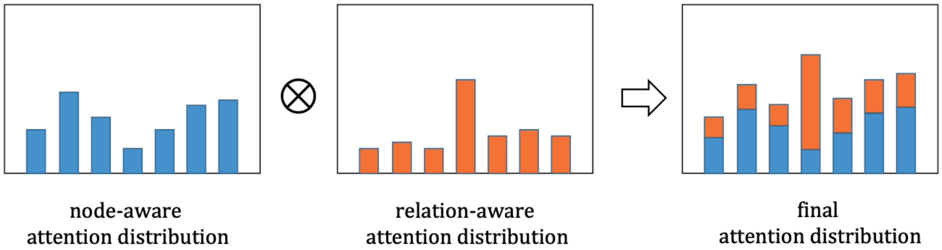

node-aware attention distribution
relation-aware attention distribution
final attention distribution

Figure 4.8: *Mixing Operation of Attentions*

For the contextual encoder, We used the BERTje model we fine-tuned for the sentiment classification in the previous method with the same experiment dataset. That means we fine-tuned the BERTje model on the manually annotated dataset and then used it as a contextual encoder in RGAT training for the manually annotated dataset experiment. The contextual encoder uses only the input text sequence as a whole for sentiment classification. It does not make use of aspect annotation or any explicit information regarding the input sequence. The RGAT encoder uses relation label embeddings and the adjacency matrix of the dependency tree to calculate two separate attentions, relation-aware attention e_{ij}^{R}, and node-aware attention e_{ij}^{N}. Then e_{ij}^{R} and e_{ij}^{N} are combined and normalized using the formula 4.2. Figure 4.8 is the illustration from the paper [3] for mixing operation of node-aware and relation-aware attentions.

$$\hat{\alpha}_{ij} = \frac{\exp\left(e_{ij}^{N} + e_{ij}^{R}\right)}{\sum_{j' \in \mathcal{N}(i)} \exp\left(e_{ij'}^{N} + e_{ij'}^{R}\right)} \tag{4.2}$$

$$h_i^l = \mathop{||}_{k=1}^{Z} \sigma\Big(\sum_{j \in \mathcal{N}(i)} \hat{\alpha}_{ij}^{lz} \left(W_V^{lz} h_j^{l-1} + W_{Vr}^{l} r_{ij} \right)\Big) \tag{4.3}$$

The combined attention weights are used to aggregate information from neighbor nodes (words) using the formula 4.3. $||$ represents vector concatenation; Z indicates the number of attention heads. σ is the sigmoid function. $\hat{\alpha}_{ij}^{lz}$ denotes the combined attention weight, W_{Vr}^{l} is the parameter matrix and finally r_{ij} describes the relation vector.

Pooling operations shown in figure 4.7 are average pooling functions. The output of the pooling operations is stated as RGAT encoding (h_s) and BERT encoding (h_c). RGAT and BERT encodings are subject to feature fusion operation. Feature fusion is an element-wise product operation with a gating mechanism. The gating mechanism g controls the fusion rate and is calculated with the formula 4.5 where ';' indicates concatenation of encodings, W_g and b_g are parameters. The equation 4.4 shows the calculation of future fusion representation h_f where $\circ$ represents element-wise product operation. The classifier is a linear layer that takes h_f as input and provides the probability distribution over negative and positive polarities.

$$h_f = g \circ h_s + (1 - g) \circ h_c \tag{4.4}$$

$$g = \sigma\left(W_g \left[h_s; h_c\right] + b_g\right) \tag{4.5}$$

Chapter 5

Results

This chapter presents the results from the models, describes the experiments, and shows the cross-hospital analysis and error analysis. We run experiments on two datasets: a small dataset and a large dataset. The small dataset is annotated with aspects manually. The large dataset is annotated with aspects using a NER model trained on the small dataset. Each experiment is performed in a 5-fold cross-validation fashion. We report the average performances of the best models in every fold.

We examined the contribution of syntactic dependency information to the polarity classification task through our experiments. We compared GraphSAGE-GCN and RGAT models with state-of-the-art BERT variations and conventional methods. In the aforementioned contribution investigation, we examined the merits of the syntactic dependency label in addition to the graph obtained only from the syntactic dependency relations.

For the assessment of the test set, we monitored accuracy and macro-averaged F1 score metrics. We reported training loss and validation loss along with the validation accuracy metric for training evaluation. Training and validation losses are calculated using the cross-entropy loss from the torch library. Since the SVM model does not have a function to provide probability distribution over polarities, we could not calculate its loss and kept it blank in the tables. The performances are sorted based on accuracy metrics.

5.1 Named Entity Recognition Results

We found that recommended settings on the Flair tutorial page[1] produce the best results for the Dutch NER model. Table 5.1 shows the performance of the NER model by class. The model performs worst in the class *Behandeling*. The diversity of annotation keywords might cause worse performances. Also, some aspects are similar such as Zorg and Behandeling or Personneel and Vriendelijkheid. For instance, the aspect Vriendelijkheid generally takes place along with the words such as doctors, nurses, etc. Since we annotate only one aspect in each sentence based on our assumption, this might make it difficult for NER model to learn effectively.

NER model training log indicates the accuracy of the NER model is 0.49. Flair uses IOBES tag scheme. The reported accuracy is low because it is calculated with a strict matching condition on all the tags, including the O tag. The table 5.1 presents metrics by class. In this presentation, O tags are discarded, and the metrics are calculated accordingly. We also

[1] https://github.com/flairNLP/flair/blob/master/resources/docs/EXPERIMENTS.md

	precision	*recall*	*f1-score*	*support*
COMMUNICATIE	0.63	0.67	0.65	79
WACHTTIJD	0.70	0.76	0.73	41
ZORG	0.58	0.74	0.65	34
SNELHEID	0.68	0.76	0.72	34
BEHANDELING	0.19	0.19	0.19	26
VRIENDELIJKHEID	0.73	0.70	0.71	23
ETEN	0.75	0.69	0.72	13
PERSONEEL	0.70	0.64	0.67	11
SCHOONMAKEN	0.78	0.70	0.74	10
micro avg	0.62	0.66	0.64	271
macro avg	0.64	0.65	0.64	271
weighted avg	0.62	0.66	0.64	271

Table 5.1: *NER Model Classification Report on Test set*

calculated the accuracy by disregarding the O tag. This calculation is also made with a strict matching condition.

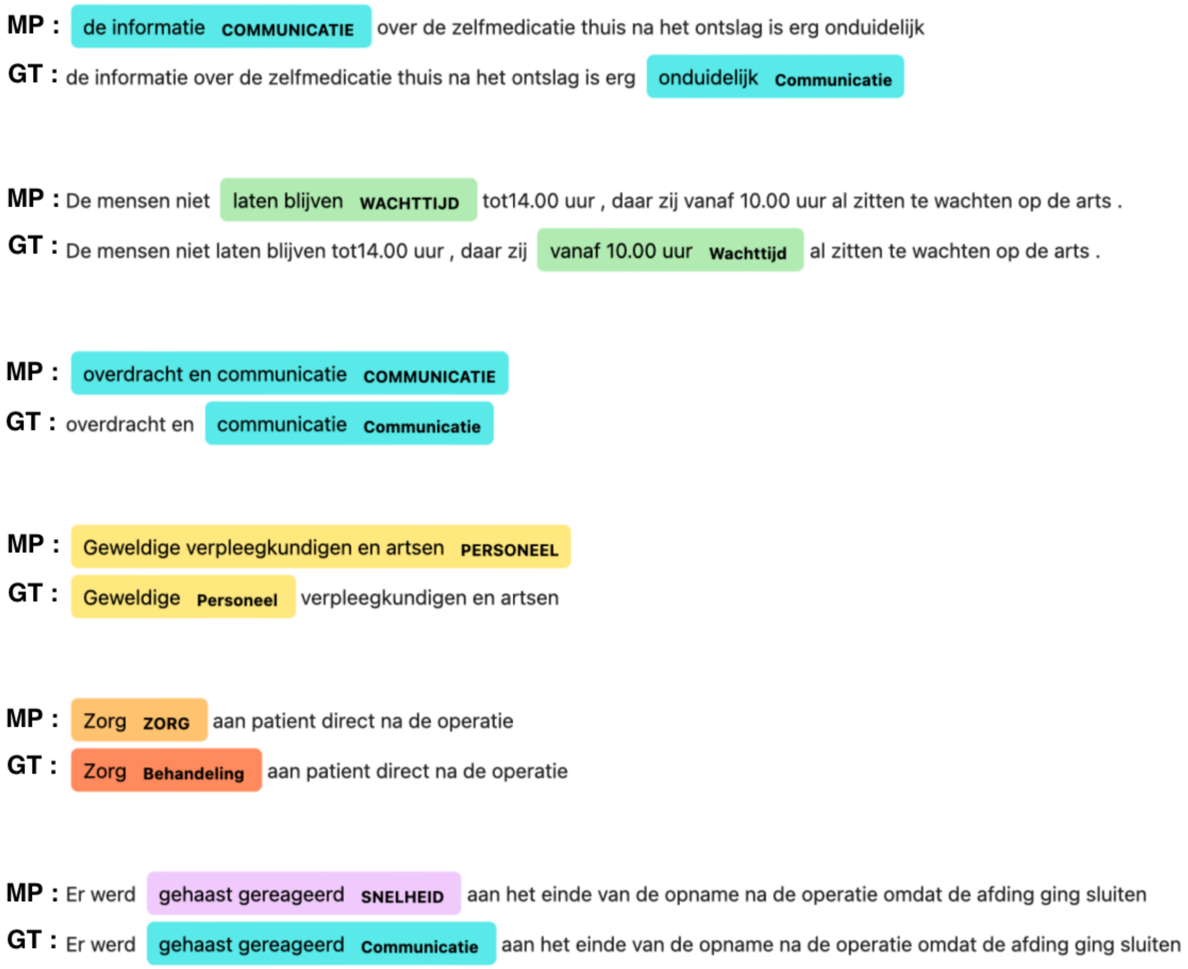
By looking at the predictions made by the model, we found that in some cases, a non-exact match could also be regarded as a valid prediction. Figure 5.1 illustrates some examples of NER predictions. MP refers to model prediction, and GT refers to ground truth. The first and second examples illustrate the correctly predicted aspect labels but non-overlapping annotations. The third and fourth examples demonstrate correctly predicted overlapping annotations. The last two examples display the wrong predictions made by the NER model. The overlapping annotations, such as the third and fourth illustrations in figure 5.1 are the examples where the model prediction encapsulates the ground truth. However, overlapping annotations were considered false in accuracy calculation. For this reason, we made two approaches to recalculate the accuracy of the NER model. The first approach considers predictions valid only if the predicted aspect is equal to ground truth and there is an overlap. The second approach only looks for the equivalence in aspect labels. So, the first and second examples in figure 5.1 accounted as true only in the second approach, while the third and fourth examples are considered valid in both methods.

With these additional calculation methods, the performance of the NER model is as follows;

- Accuracy (Training log) : 0.49
- Accuracy (Without 'O' tag) : 0.66
- Accuracy (Label with overlap) : 0.79
- Accuracy (Label) : 0.88

5.2 Manually Annotated Dataset Results

Table 5.2 shows the test set results of small dataset. The training and validation performances are presented in table 8.9 in Appendix C. In the small dataset, the conventional



MP : de informatie **COMMUNICATIE** over de zelfmedicatie thuis na het ontslag is erg onduidelijk

GT : de informatie over de zelfmedicatie thuis na het ontslag is erg onduidelijk **Communicatie**

MP : De mensen niet laten blijven **WACHTTIJD** tot14.00 uur , daar zij vanaf 10.00 uur al zitten te wachten op de arts .

GT : De mensen niet laten blijven tot14.00 uur , daar zij vanaf 10.00 uur **Wachttijd** al zitten te wachten op de arts .

MP : overdracht en communicatie **COMMUNICATIE**

GT : overdracht en communicatie **Communicatie**

MP : Geweldige verpleegkundigen en artsen **PERSONEEL**

GT : Geweldige **Personeel** verpleegkundigen en artsen

MP : Zorg **ZORG** aan patient direct na de operatie

GT : Zorg **Behandeling** aan patient direct na de operatie

MP : Er werd gehaast gereageerd **SNELHEID** aan het einde van de opname na de operatie omdat de afding ging sluiten

GT : Er werd gehaast gereageerd **Communicatie** aan het einde van de opname na de operatie omdat de afding ging sluiten

Figure 5.1: *NER Model Prediction Examples*

methods SVM and Logistic Regression performed better than RobBERT and GCN neural networks despite having a greater loss in training and validation. BERTje and mBERT models achieved more promising results with a small margin. The loss values of the RGAT model show that integrating dependency information significantly improves the training results. However, the RGAT model suffered the biggest decrease in the accuracy in the table 5.2. Only the conventional methods, Logistic Regression, Naive Bayes, and Random Forest, performed better in the test set.

5.3 Automatically Annotated Dataset Results

Table 5.3 presents the test performances for large dataset. Training and validation performances are shown in table 8.10 in Appendix C. Neural network architectures are data-hungry and large dataset contributes to their success in generalization. On the other hand, the conventional methods, such as SVM, Logistic Regression, etc., are expected to scale poorly with large datasets due to computational costs or worse estimation [17][13]. We confirmed that neural networks are better in scaling because the best conventional approaches like Logistic Regression and SVM performed fairly well in the small test set, while the neural network approaches except RGAT method performed worse probably due to underfitting. Still, BERT-

Rank	*Method*	*Acc. (%)* ↑	*F1-score*
1	***RGAT***	**94.50**	**0.9453**
2	*SVM*	92.84	0.9282
3	*Logistic Regression*	92.84	0.9281
4	*BERTje*	92.69	0.9209
5	*Naive Bayes*	92.03	0.9199
6	*Random Forest*	91.37	0.9134
7	*RobBERT_v2*	90.34	0.8964
8	*mBERT*	90.33	0.8974
9	*GS-GCN*	90.26	0.9023
10	*Ada Boost*	86.94	0.8686
11	*Decision Tree*	81.62	0.8161
12	*Dictionary-Based*	45.90	0.3139

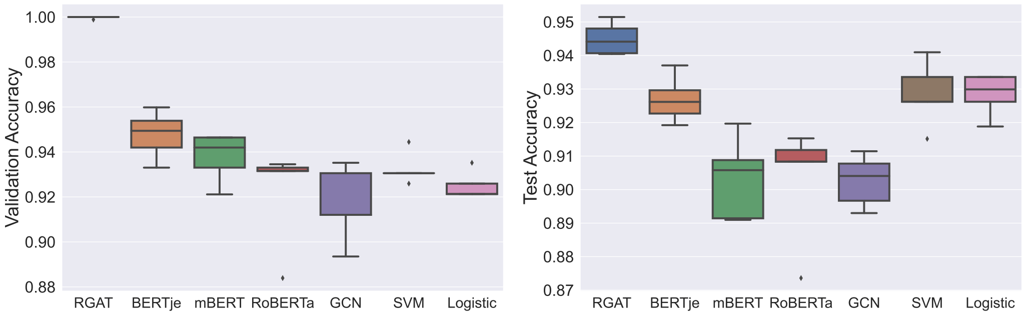
Table 5.2: *Manually Annotated Test Set Performances*

Rank	*Method*	*Acc. (%)* ↑	*F1-score*
1	***BERTje***	**88.81**	**0.8845**
2	*RGAT*	88.74	0.8867
3	*mBERT*	87.82	0.8743
4	*Logistic Regression*	85.41	0.8540
5	*SVM*	85.40	0.8536
6	*Naive Bayes*	84.15	0.8411
7	*Random Forest*	83.69	0.8368
8	*GS-GCN*	80.08	0.7971
9	*Decision Tree*	77.12	0.7700
10	*Ada Boost*	76.80	0.7630
11	*Dictionary-Based*	47.79	0.3250

Table 5.3: *Automatically Annotated Test Set Performances*

based approaches achieved better accuracy in the large dataset. The performance boost in the RGAT model was smaller in the large training set. Even though the F1-score of the RGAT model was slightly higher than the BERTje model, it took second place in table 5.3.

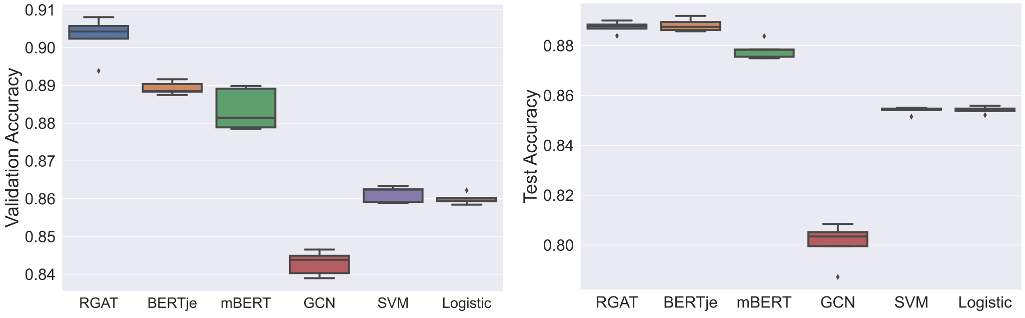
In order to see how consistent the results are across different folds, we used box plots. Random Forest, Ada Boost, and Decision Tree models have not been included in box plot representations. Figures 5.2 and 5.3 display the performances across 5-folds for small and large datasets respectively. In the small dataset, mBERT and GraphSAGE-GCN models' performances have a wider interval than other approaches. In the large test set, even though the BERTje model was ranked as the first based on average accuracy RGAT model seems to have a more consistent accuracy.



(a) *Validation Set*

(b) *Test Set*

Figure 5.2: *Manually Annotated Dataset Performances*



(a) *Validation Set*

(b) *Test Set*

Figure 5.3: *Automatically Annotated Dataset Performances*

5.4 Trainable Parameter Comparison

When the performances are considered, we see that the GS-GCN model achieves far less accuracy than other neural network models. This can be explained by the differences in the depth of the models. Table 5.4 presents trainable parameter counts for each neural network we employed. GS-GCN model is 828 times smaller than BERTje, 1270 times smaller than mBERT model. So, the performance gap is a natural by-product of this difference.

5.5 Cross-Hospital Analysis

The models are trained on the automatically annotated large dataset from hospitals 1 and 2. We evaluated them on the automatically annotated hospital 3 dataset. Table 5.5 shows the performances. The ranking of the models did not change much. However, BERTje and mBERT models got poor f1 scores.

5.6 Error Analysis

In this section, we automatically labeled the large dataset from Hospital 1&2 and the dataset from Hospital 3 using the BERTje model trained on the large dataset. We demonstrate some of the tables and figures in Appendix D to increase the readability of this section. The model predicts based on the maximum probability of a sentiment. In order to investigate

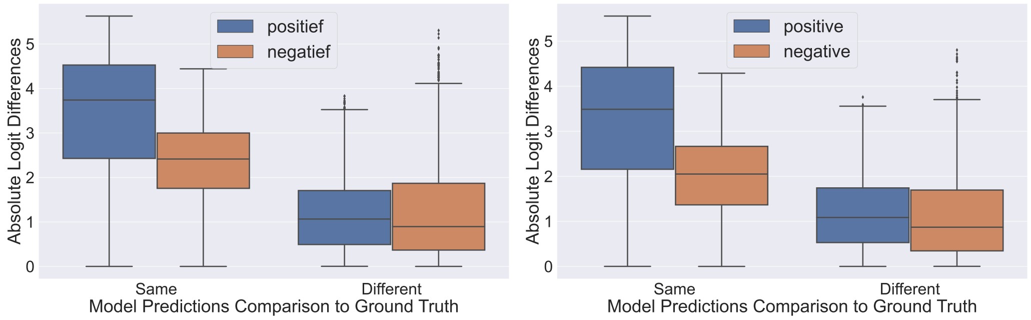
Method	***Number of Parameters***
mBERT	167,357,954
RobBERT_v2	116,763,650
RGAT	109,638,022
BERTje	109,138,946
GS-GCN	131,675

Table 5.4: *Trainable Parameter Comparison*

Rank	***Method***	***Acc. (%) ↑***	***F1-score***
1	***RGAT***	**87.34**	**0.8714**
2	*BERTje*	87.34	0.4832
3	*mBERT*	86.45	0.4730
4	*Logistic Regression*	84.23	0.8404
5	*SVM*	84.28	0.8402
6	*Naive Bayes*	82.91	0.8269
7	*Random Forest*	82.47	0.8233
8	*GS-GCN*	82.28	0.8212
9	*Decision Tree*	76.39	0.7601
10	*Ada Boost*	75.68	0.7475

Table 5.5: *Automatically Annotated Hospital 3 Dataset Performances*

the confidence of the wrongfully predicted responses, we recorded the logit values and calculated the absolute difference between prediction labels. The tables 8.11 and 8.12 present the distribution of confidence values, and figure 5.4 displays them as a box-plot. In both datasets, the average confidence of wrongly predicted responses is lower than correctly predicted ones.



(a) *Hospital 1&2*

(b) *Hospital 3*

Figure 5.4: *Distribution of prediction confidence*

We investigated this further and checked the top-10 misclassified responses. Tables 8.13-8.14 and 8.15-8.16 in Appendix D exemplify the misclassification cases for the large dataset and Hospital 3 dataset. Misclassified cases show that the datasets are quite noisy when just labeled based on the columns. The BERTje model predicts the sentiments correctly; however, it is marked as a false prediction due to the noise in the dataset. This situation shows that the performances presented for the large and H3 datasets are underestimated. It also exhibits that sentiment analysis is essential even if the questionnaire questions clearly display the sentiment for the patients to fill in.

We looked at the attention weights for the falsely predicted sentences for deeper error analysis. Even for a human annotator, deciding the comments' polarity could be challenging. From the table 8.15 we know that the ground truth of the misclassified comments mainly indicates the wrong sentiment. We selected examples from Hospital 3 dataset with the correct ground truth but are misclassified by the BERTje model with high confidence. This selection process was also challenging because the ground truths were not obvious. We selected them in the context of the questions: "What went well?" and "What could be improved?". The table 8.17 presents those examples. The examples do not indicate a sentiment since they mainly answer the questions that include those sentiments. When the comments are thought in the context of the questions, it is easier to classify them. This also shows that the questionnaire should be carefully designed to get proper answers.

The figure 8.7 illustrates the tokens with the highest attention weights for the classification for some examples. The model has 11 attention layers and heads. We selected the layer that shows the clear distinction for tokens. Attention figure uses blue and orange color tones to show positive and negative attention weights respectively. The darker the color the higher the value in corresponding token weights. Figures illustrate the path to the token that contributes the most to the classification. Another problem seen in figure 8.7c is the tokenization of the BERT model. This also might adversely affect the performance.

We also looked at a few misclassified examples with the lowest confidence. Table 8.18 shows the selected examples, and figure 8.8 displays the tokens with the highest attention value. In all the examples except the fourth, the BERTje model gave low attention values to the aspect sequence which contains important information about the sentiment. The aspect words contain important information about the sentiment. The model should focus on aspects and related words to improve classification performance. Providing explicit information, such as dependency relations, as in the RGAT approach, helps the model pay more attention to the aspects, hence assists sentiment classification performance.

Chapter 6

Discussion & Future Work

6.1 Challenges

In this work, we attempted to see the benefits of the syntactic dependency information to solve an aspect-based sentiment classification task in the healthcare domain. The main challenges are preprocessing the text and the lack of annotated data. Preprocessing becomes a challenge due to domain-specific and hospital-specific abbreviations such as ”o.k”: ”operatie kamer”, ”pa”: ”psychiatrische afdeling”, and ”C11”: ”a building in Hospital 1”.

Handling preprocessing steps with the help of domain experts took too much time. Also, reaching a high agreement in annotations is quite hard for human annotators, especially regarding the annotation of compound words. In addition, the data was noisy: the negative column of the dataset does not represent negative sentiment. Non-representative negative texts cause too much noise in the data. These problems make the annotation less reliable and cause a performance decrease in models.

6.2 Comparison of Models

In both small and large training sets RGAT model performed the best. It was expected because we incorporated the BERTje model trained on the same dataset as the contextual encoder in the RGAT model architecture. The performance boost in RGAT over BERTje clearly shows the contribution of integrating dependency information explicitly. BERT models learn such information implicitly; however, the RGAT approach proved the benefits of explicit integration of syntactic dependencies on other domains. Our experiment results also acknowledge its capabilities in the medical data.

For BERT models, even though the RobBERT version 2 performs the best on most downstream tasks[11], in our medical dataset, the best model was BERTje, and RobBERT performed the worst among the three BERT models. For this reason, and due to the computation cost, we did not utilize the RobBERT model on the large dataset.

6.3 Limitations

BERT models implicitly learn syntactic information such as dependency information and position embeddings. Explicit integration of syntactic information enriches the representation;

however, according to Bai *et al.* [3] fine-grained syntactic info creates a more sparse discrete structure which makes learning the proper representation harder for the model.

Extracting discrete structures also causes inconsistency. For example, in order to extract dependency relations, we employed the Stanza library. Stanza library uses a pipeline to extract features. The tokenization step is not always the same as the BERT tokenizer. For this reason, the output dependency tree and tokens are sometimes not the same length or do not indicate the correct tokens.

Both GS-GCN and RGAT models operate on one sentence. Thus these models are limited to sentence-level analysis. Naturally, the original dataset includes multi-sentence comments. To work with these models, we split the sentences while keeping the original label for each. Thus, an information loss occurred because not all sentences include a sentiment, but the combination of these models does. Also, in these approaches, we made a naive assumption that each sentence contains only one aspect. In reality, this is not the case. Sentences are labeled as either positive or negative. However, it is common for survey data to present positive and negative sentiments in the same sentence. For better sentiment classification, more fine-grained models are needed to address such problems.

Another limitation is the separate annotations for negative and positive aspects. We selected different aspects to be annotated based on the frequency encountered during topic modeling and cleaning the data. More comprehensive aspect selection might be better for custom NER model training; thus less noisy dataset could be created with such a model.

6.4 Practical Application/Utility for the hospitals

The hospitals require aspects from the survey data to identify the sentiments over different topics, take necessary steps to improve them, and keep track of the progress over time. With our research, we provided a pipeline to identify aspects using NER predictions extracting the sentiments over them using a sentiment analysis model. For the end-user (doctors, department managers), our work provides valuable insight that helps them take action on the aspects.

6.5 Future Work

With the topic modeling part of our research, we showed the challenges regarding the survey questions, which led the hospitals to prepare an unequivocal survey. Recently, with multiple hospitals' contributions, a new survey called AI-PREM has been prepared to address the challenges seen in multiple studies, including ours.

More comprehensive explanations are needed to provide more rigorous models. We left making large-scale annotations with the agreement of more than one commentator for future studies.

Chapter 7

Conclusion

In this study, we examined the benefits of syntactic dependencies on aspect-based sentiment analysis in the healthcare domain for the Dutch language. We compared neural network approaches that integrate dependency relations in their architecture with state-of-the-art BERT-based models and conventional methods such as SVM, Logistic Regression, and Naive Bayes. We performed a cross-domain analysis for the RGAT model. Our experiments showed that integrating syntactic relations into text representation improves the aspect-based sentiment analysis result. We extracted the positive and negative aspects of three hospitals' data and compared them. The cross-hospital analysis demonstrated the transferability of our aspect annotations and model performances. We showed that explicit integration of syntactic dependencies is a viable approach to uncover the strengths and weaknesses of hospital care. Our research provided actionable insight for hospitals to improve their healthcare services.

Bibliography

[1] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. Flair: An easy-to-use framework for state-of-the-art nlp. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59, 2019.

[2] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics.

[3] Xuefeng Bai, Pengbo Liu, and Yue Zhang. Investigating typed syntactic dependencies for targeted sentiment classification using graph attention neural network. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:503–514, 2020.

[4] Lingxian Bao, Patrik Lambert, and Toni Badia. Attention and lexicon regularized lstm for aspect-based sentiment analysis. In *Proceedings of the 57th annual meeting of the association for computational linguistics: student research workshop*, pages 253–259, 2019.

[5] Carla M Bastemeijer, Hileen Boosman, Linda Zandbelt, Reinier Timman, Dolf de Boer, and Jan A Hazelzet. Patient experience monitor (pem): the development of new short-form picker experience questionnaires for hospital patients with a wide range of literacy levels. *Patient Related Outcome Measures*, 11:221, 2020.

[6] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022, 2003.

[7] Simone A Cammel, Marit S De Vos, Daphne van Soest, Kristina M Hettne, Fred Boer, Ewout W Steyerberg, and Hileen Boosman. How to automatically turn patient experience free-text responses into actionable insights: a natural language programming (nlp) approach. *BMC medical informatics and decision making*, 20(1):1–10, 2020.

[8] Orphée De Clercq and Véronique Hoste. Rude waiter but mouthwatering pastries! an exploratory study into dutch aspect-based sentiment analysis. In *10th International Conference on Language Resources and Evaluation (LREC)*, pages 2910–2917. ELRA, 2016.

[9] Orphée De Clercq, Els Lefever, Gilles Jacobs, Tijl Carpels, and Véronique Hoste. Towards an integrated pipeline for aspect-based sentiment analysis in various domains. In *Proceedings of the 8th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 136–142, 2017.

[10] Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. Bertje: A dutch bert model. *arXiv preprint arXiv:1912.09582*, 2019.

[11] Pieter Delobelle, Thomas Winters, and Bettina Berendt. Robbert: a dutch roberta-based language model. *arXiv preprint arXiv:2001.06286*, 2020.

[12] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

[13] Aurélien Géron. *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*, pages 100–102. "O'Reilly Media, Inc.", 2019.

[14] Felix Gräßer, Surya Kallumadi, Hagen Malberg, and Sebastian Zaunseder. Aspect-based sentiment analysis of drug reviews applying cross-domain and cross-data learning. In *Proceedings of the 2018 International Conference on Digital Health*, pages 121–125, 2018.

[15] Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[16] Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

[17] Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. *The elements of statistical learning: data mining, inference, and prediction*, volume 2, page 455. Springer, 2009.

[18] Mickel Hoang, Oskar Alija Bihorac, and Jacobo Rouces. Aspect-based sentiment analysis using bert. In *Proceedings of the 22nd nordic conference on computational linguistics*, pages 187–196, 2019.

[19] Binxuan Huang and Kathleen M Carley. Syntax-aware aspect level sentiment classification with graph attention networks. *arXiv preprint arXiv:1909.02606*, 2019.

[20] Daniel D Lee and H Sebastian Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–791, 1999.

[21] Andrew Kachites McCallum. Mallet: A machine learning for language toolkit. http://mallet.cs.umass.edu, 2002.

[22] Tomas Mikolov, Edouard Grave, Piotr Bojanowski, Christian Puhrsch, and Armand Joulin. Advances in pre-training distributed word representations. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[23] László Nemes and Attila Kiss. Information extraction and named entity recognition supported social media sentiment analysis during the covid-19 pandemic. *Applied Sciences*, 11(22):11017, 2021.

[24] Sebastian Ruder, Parsa Ghaffari, and John G Breslin. A hierarchical model of reviews for aspect-based sentiment analysis. *arXiv preprint arXiv:1609.02745*, 2016.

[25] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.

[26] Chi Sun, Luyao Huang, and Xipeng Qiu. Utilizing bert for aspect-based sentiment analysis via constructing auxiliary sentence. *arXiv preprint arXiv:1903.09588*, 2019.

[27] Kai Sun, Richong Zhang, Samuel Mensah, Yongyi Mao, and Xudong Liu. Aspect-level sentiment analysis via convolution over dependency tree. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pages 5679–5688, 2019.

[28] Yequan Wang, Minlie Huang, Xiaoyan Zhu, and Li Zhao. Attention-based lstm for aspect-level sentiment classification. In *Proceedings of the 2016 conference on empirical methods in natural language processing*, pages 606–615, 2016.

[29] Hu Xu, Bing Liu, Lei Shu, and Philip S Yu. Bert post-training for review reading comprehension and aspect-based sentiment analysis. *arXiv preprint arXiv:1904.02232*, 2019.

[30] Chen Zhang, Qiuchi Li, and Dawei Song. Aspect-based sentiment classification with aspect-specific graph convolutional networks. *arXiv preprint arXiv:1909.03477*, 2019.

[31] Anastazia Žunić, Padraig Corcoran, and Irena Spasić. Aspect-based sentiment analysis with graph convolution over syntactic dependencies. *Artificial Intelligence in Medicine*, 119:102138, 2021.

[32] Anastazia Zunic, Padraig Corcoran, Irena Spasic, et al. Sentiment analysis in health and well-being: systematic review. *JMIR medical informatics*, 8(1):e16023, 2020.

Chapter 8

Appendices

A Topic Modeling Analysis

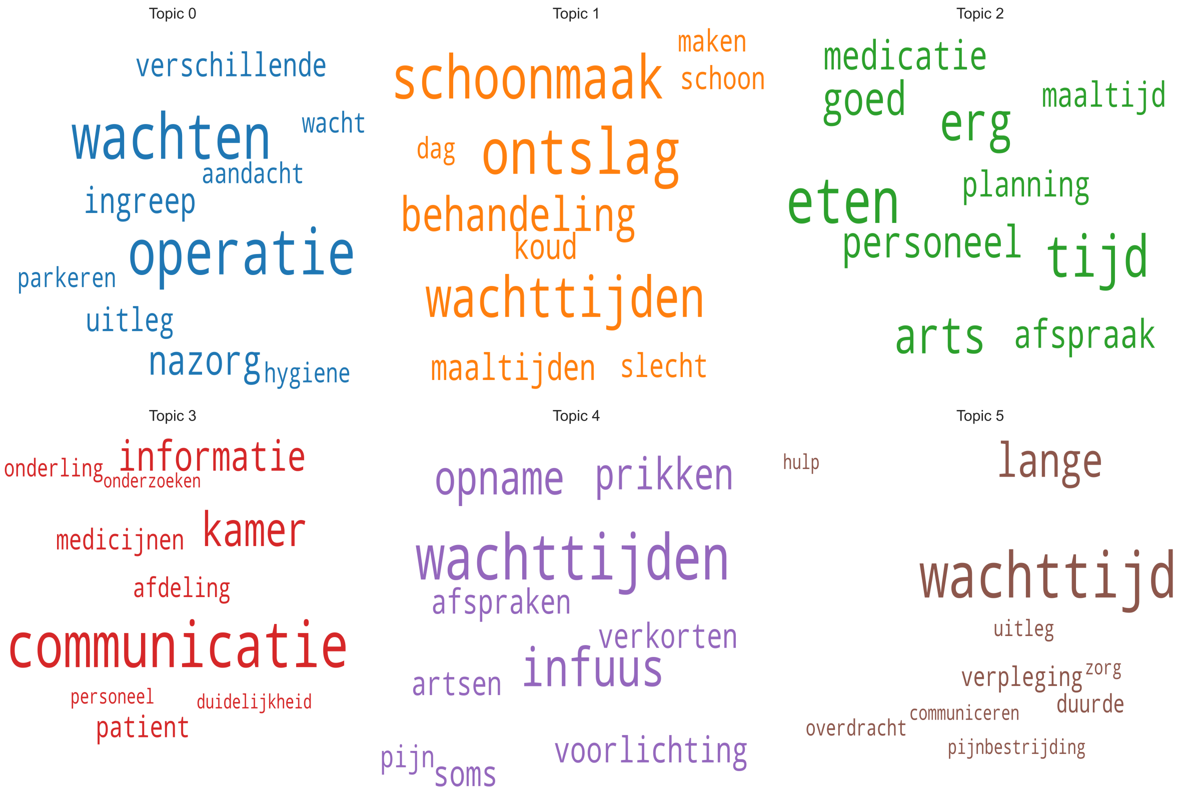


Figure 8.1: *Top-10 Words per Topic in Negative Responses*

Topic Num	*Topic Perc. Contrib.*	*Keywords*	*Example Comment*
0	0.2138	operatie, wachten, nazorg, ingreep, uitleg, ver...	De wacht tijden veranderen
1	0.2202	ontslag, schoonmaak, wachttijden, behandeling, ...	overdragen hulpmiddelen zorgvuldiger werken
2	0.2138	eten, tijd, erg, arts, goed, personeel, medicat...	Het voelde ietwat onpersoonlijk
3	0.2138	communicatie, kamer, informatie, patient, medic...	Vinden van de juiste lift
4	0.2099	wachttijden, infuus, opname, prikken, voorlicht...	Vervolg traject uitgebreider bespreken
5	0.2138	wachttijd, lange, verpleging, duurde, uitleg, o...	Lange wachttijd na bevalling

Table 8.1: *The negative documents which has the highest percentage contribution in each topic*

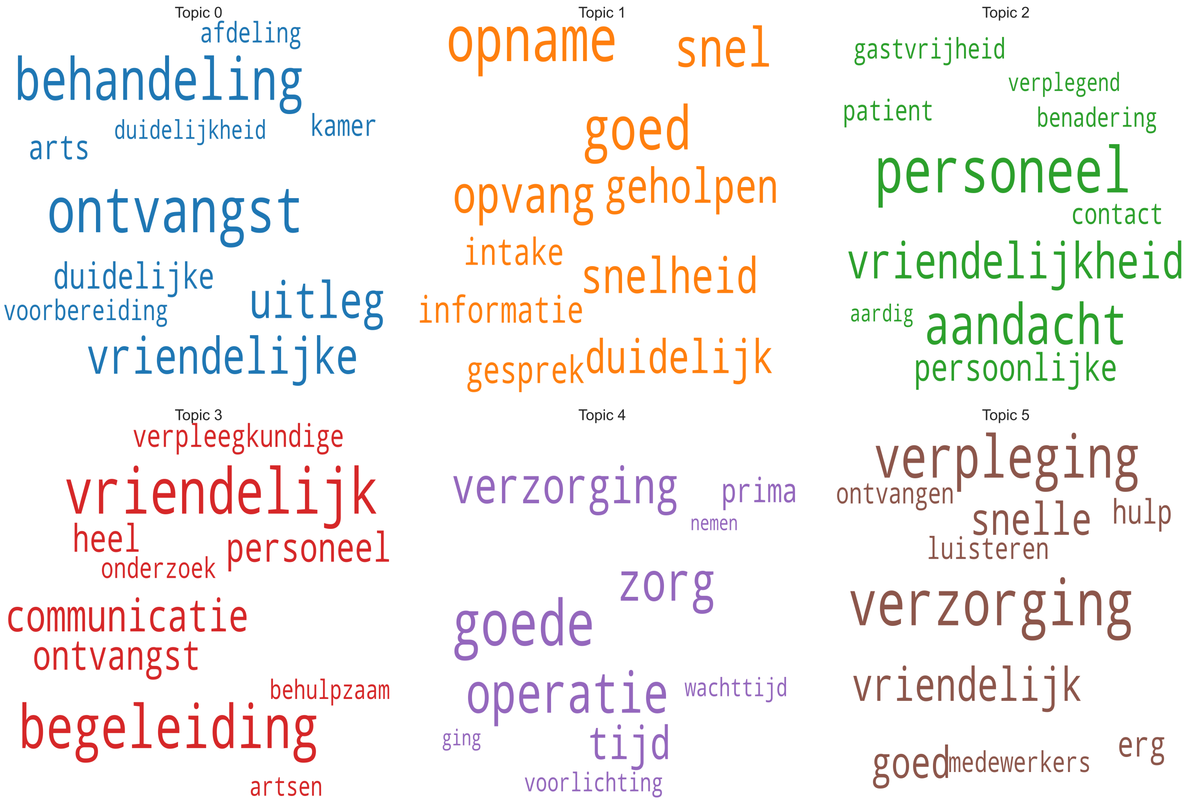


Figure 8.2: *Top-10 Words per Topic in Positive Responses*

Topic Num	*Topic Perc. Contrib.*	*Keywords*	*Example Comment*
0	0.2284	ontvangst, behandeling, uitleg, vriendelijke, d...	duidelijke uitleg en deskundige mensen
1	0.2222	opname, goed, snel, opvang, snelheid, geholpen,...	Goed zichtbaar Duidelijk uitslag zichtbaar
2	0.2284	personeel, aandacht, vriendelijkheid, persoonli...	opvallend meelevend en deskundig personeel
3	0.2162	vriendelijk, begeleiding, communicatie, ontvang...	heel snel start grondig onderzoek
4	0.2284	goede, operatie, zorg, verzorging, tijd, prima,...	fijn en open transplantatie team
5	0.2284	verzorging, verpleging, vriendelijk, snelle, go...	snelle hulp bij spoedeisende hulp

Table 8.2: *The positive documents which has the highest percentage contribution in each topic*

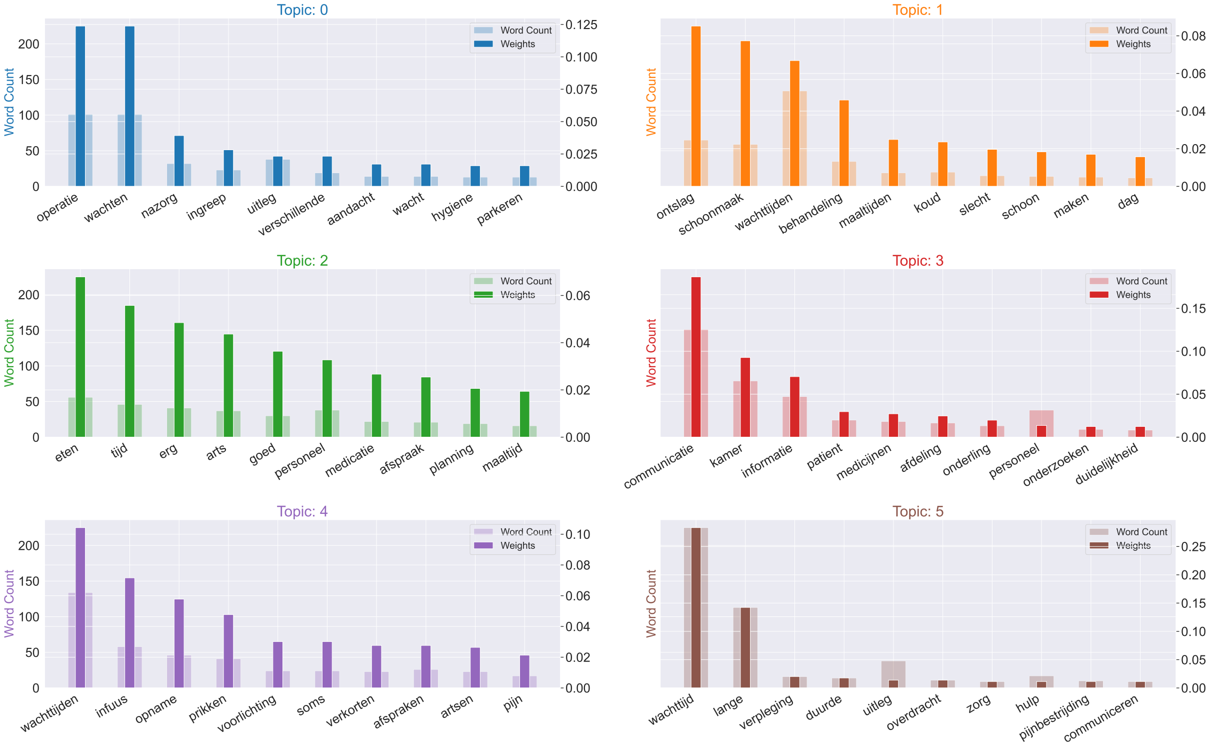
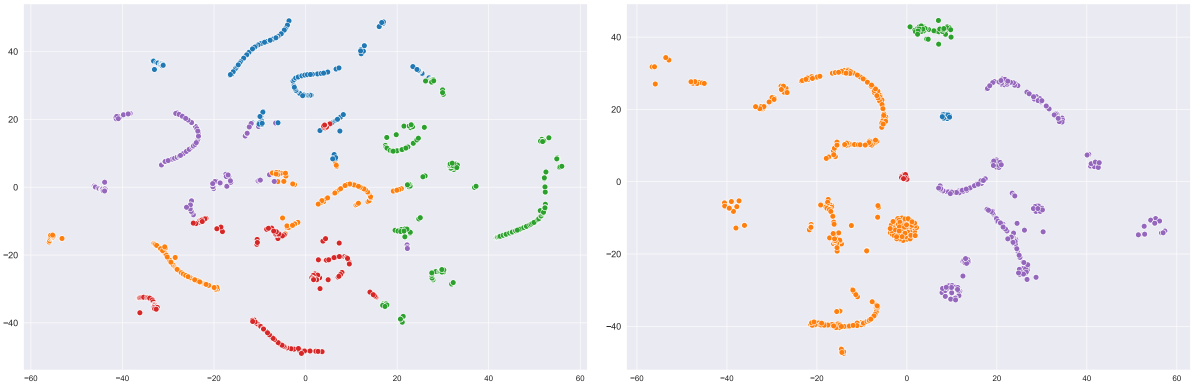


Figure 8.3: *Word Count and Importance of Topic Keywords in Negative Responses*



(a) *Negative Responses*

(b) *Positive Responses*

Figure 8.4: *t-SNE Clustering*

Topic 1	***Topic 2***	***Topic 3***	***Topic 4***	***Topic 5***
goede verzorging behandeling verzorging verzorging verpleging verzorging personeel algehele verzorging	ging goed heel goed goed opgevangen ging prima goed	vriendelijke ontvangst ontvangst begeleiding goede ontvangst ontvangst afdeling ontvangst opname	voorbereiding operatie operatie kamer begeleiding operatie uitleg operatie verzorging operatie	vriendelijk personeel zeer vriendelijk heel vriendelijk vriendelijk behulpzaam vriendelijk
Topic 6	***Topic 7***	***Topic 8***	***Topic 9***	***Topic 10***
goede begeleiding ontvangst begeleiding opvang begeleiding begeleiding personeel begeleiding operatie	persoonlijke aandacht aandacht patiënt aandacht personeel zorg aandacht verzorging aandacht	vriendelijkheid personeel vriendelijkheid verpleging vriendelijkheid medewerkers vriendelijkheid verpleegkundigen vriendelijkheid	snelle behandeling behandeling verzorging vriendelijke behandeling behandeling opname	goede zorg zorg aandacht zorg personeel zorg verpleging goede verzorging

Table 8.3: *The top-5 positive phrases for each topic identified by NMF algorithm*

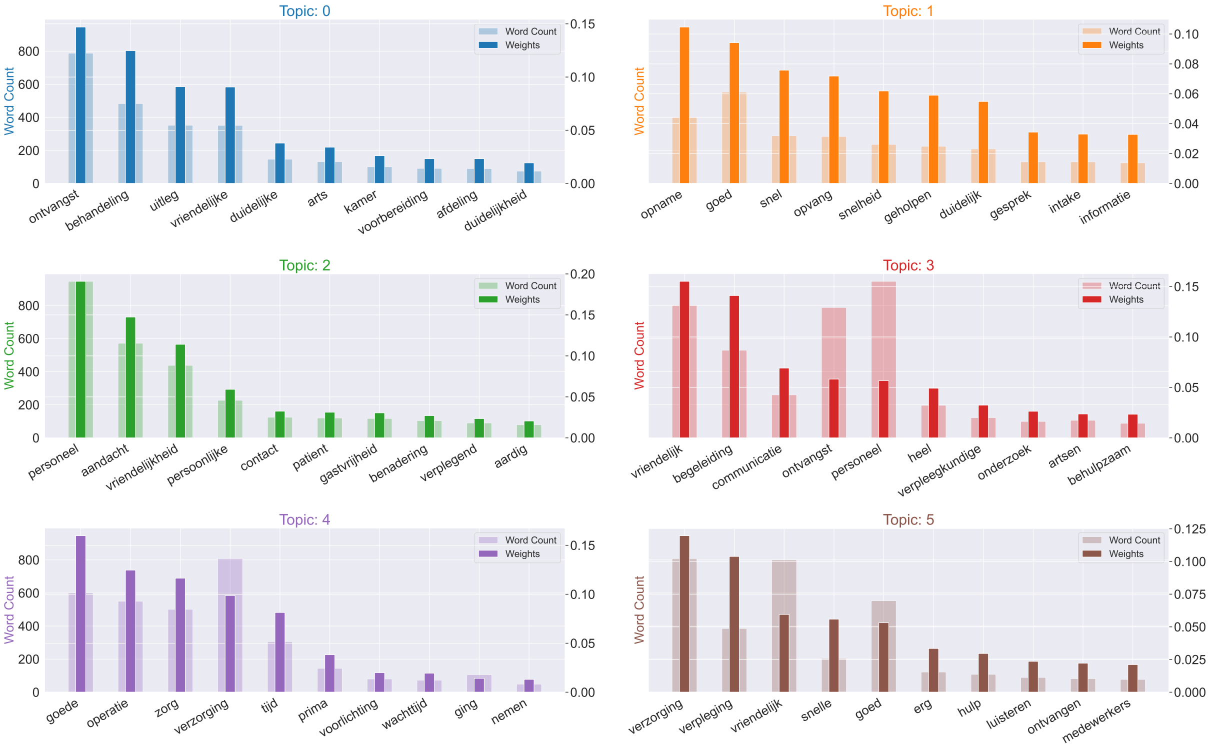
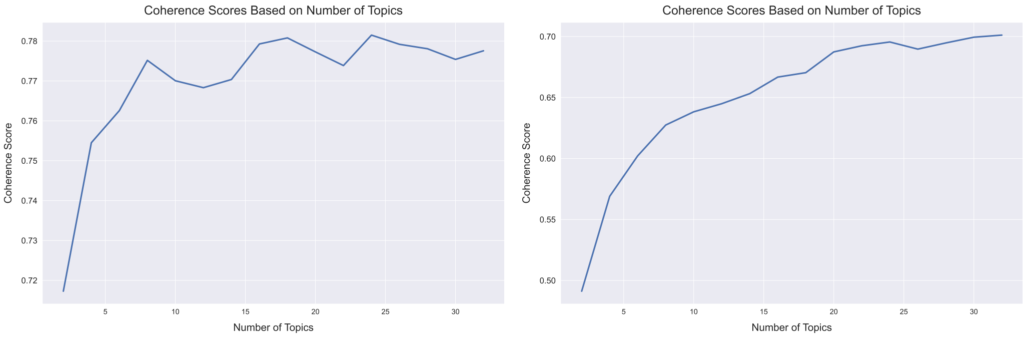


Figure 8.5: *Word Count and Importance of Topic Keywords in Positive Responses*



(a) *Negative Responses*

(b) *Positive Responses*

Figure 8.6: *Coherence Scores for Various Topic Numbers*

Topic 1	***Topic 2***
lange wachttijd	communicatie tussen
wachttijd operatie	communicatie personeel
wachttijd lang	betere communicatie
lange wachttijden	communicatie onderling
wachttijd	communicatie

Table 8.4: *The top-5 negative phrases for each topic identified by NMF algorithm*

B Aspect Annotation

	1-word	***2-words***	***3-words***
1	niets	geen opmerking	niet van toepassing
2	geen	geen klacht	alles gaan goed
3	niks	geen idee	alles was goed
4	wachttijd	geen voorbeeld	gaan zo door
5	communicatie	eigenlijk niets	alles gaan prima
6	schoonmaak	lang wachttijd	weten ik niet
7	niet	alles goed	kan niets bedenken
8	zien	geen probleem	geen negatief ervaring
9	nazorg	geen aanmerking	het gaan goed
10	goed	lang wachten	ik ben tevreden

Table 8.5: *Top-10 Negative Comments*

	1-word	***2-words***	***3-words***
1	alles	de operatie	alles gaan goed
2	verzorging	vriendelijk personeel	niet van toepassing
3	ontvangst	de verzorging	alles was goed
4	vriendelijkheid	de ontvangst	snelheid van handelen
5	operatie	persoonlijk aandacht	alles gaan prima
6	communicatie	de begeleiding	de operatie zelf
7	aandacht	goed verzorging	zeer vriendelijk personeel
8	begeleiding	vriendelijk ontvangst	ontvangst en begeleiding
9	gastvrijheid	de behandeling	behandeling en verzorging
10	niets	goed begeleiding	op tijd helpen

Table 8.6: *Top-10 Positive Comments*

	1-word	***2-words***	***3-words***
1	wachttijd	lang wachttijd	de lang wachttijd
2	communicatie	wachttijd verkorten	alles operatie kamer
3	schoonmaak	de wachttijd	het lang wachten
4	zien	het ontslag	soms lang wachten
5	nazorg	infuus prikken	op tijd beginnen
6	ontslag	het eten	de wachttijd verkorten
7	weinig	het wachten	de warm maaltijd
8	net	schoonmaak kamer	communicatie tussen afdeling
9	hygiëne	de communicatie	te lang wachttijd
10	maaltijd	kort wachttijd	wachttijd erg lang

Table 8.7: *Top-10 Negative Comments After Removing Meaningless Comments*

	1-word	***2-words***	***3-words***
1	verzorging	de operatie	snelheid van handelen
2	ontvangst	vriendelijk personeel	de operatie zelf
3	vriendelijkheid	de verzorging	zeer vriendelijk personeel
4	operatie	de ontvangst	ontvangst en begeleiding
5	communicatie	persoonlijk aandacht	behandeling en verzorging
6	aandacht	de begeleiding	op tijd helpen
7	begeleiding	goed verzorging	de opname zelf
8	gastvrijheid	vriendelijk ontvangst	vriendelijk en duidelijk
9	verpleging	de behandeling	vriendelijk en behulpzaam
10	vriendelijk	goed begeleiding	alles op tijd

Table 8.8: *Top-10 Positive Comments After Removing Meaningless Comments*

C Model Performances

Rank	***Method***	***Train Loss***	***Val. Loss***	***Val. Acc. ↑***
1	***RGAT***	**0.002**	**0.00048**	**99.98**
2	*BERTje*	0.0801	0.1560	94.76
3	*mBERT*	0.1627	0.1961	93.78
4	*SVM*	-	-	93.24
5	*Logistic Regression*	0.4782	0.4976	92.50
6	*RobBERT_v2*	0.1830	0.2044	92.29
7	*Naive Bayes*	0.4373	0.4738	91.94
8	*GS-GCN*	0.1958	0.2502	91.67
9	*Random Forest*	0.3632	0.4732	89.81
10	*Ada Boost*	0.6095	0.6130	86.94
11	*Decision Tree*	0.3141	0.4969	81.67
12	*Dictionary-Based*	0.7654	0.7731	46.11

Table 8.9: *Manually Annotated Training Set (5 Fold Cross Validation Performances)*

Rank	***Method***	***Train Loss***	***Val. Loss***	***Val. Acc. ↑***
1	***RGAT***	**0.2657**	**0.2487**	**90.28**
2	*BERTje*	0.3171	0.2799	88.92
3	*mBERT*	0.2977	0.3001	88.35
4	*SVM*	-	-	86.13
5	*Logistic Regression*	0.4701	0.4831	86.01
6	*Naive Bayes*	0.4867	0.5030	84.59
7	*GS-GCN*	0.3466	0.3743	84.29
8	*Random Forest*	0.3746	0.5085	84.20
9	*Decision Tree*	0.3172	0.5340	77.88
10	*Ada Boost*	0.6860	0.6861	76.72
11	*Dictionary-Based*	0.7914	0.7910	47.78

Table 8.10: *Automatically Annotated Training Set (5 Fold Cross Validation) Performances*

D Error Analysis

Predictions	***mean***	***std***	***min***	***25%***	***50%***	***75%***	***max***
Different	1.2040	0.9325	0.0004	0.4499	1.0240	1.7294	5.3043
Same	2.8856	1.2931	0.0001	1.9589	2.8643	3.8722	5.6281

Table 8.11: *Hospital 1 & 2 Dataset Prediction Confidence*

Predictions	***mean***	***std***	***min***	***25%***	***50%***	***75%***	***max***
Different	1.2162	0.9325	0.0007	0.4724	1.0231	1.7382	4.8034
Same	2.6649	1.3696	0.0004	1.6118	2.5985	3.7269	5.5577

Table 8.12: *Hospital 3 Dataset Prediction Confidence*

Text	***Aspect***	***True***	***Pred.***	***Logits***
de behandeling ging vrij snel	Snelheid	neg	pos	4.803
Zij stelde duidelijke vragen en gaf mij het gevoel dat ik gehoord werd	Communicatie	neg	pos	4.713
Klant vriendelijk van balie personeel	Vriendelijkheid	neg	pos	4.644
Korte wachttijd	Wachttijd	neg	pos	4.616
De persoon aan de telefoon was correct en duidelijk	Communicatie	neg	pos	4.589
goede uitleg zeer belangrijk	Communicatie	neg	pos	4.552
Korte wachttijden	Wachttijd	neg	pos	4.552
Ik was erg vroeg op de afdeling en werd direct voorbereid op de operatie kamer	Snelheid	neg	pos	4.305
Begeleiding voor de ingreep en ook erna	Zorg	neg	pos	4.298
Uitleg , oveleg en nazorg	Communicatie	neg	pos	4.242

Table 8.13: *H3 Dataset Top-10 Misclassified Responses with Highest Confidence*

Text	***Aspect***	***True***	***Pred.***	***Logits***
De organisatie in deze lastige Corona periode	Zorg	pos	neg	0.000655
Op zich aardig en vriendelijk [datum] weinig ingaan op de problemen	Vriendelijkheid	pos	neg	0.002146
Ik vond de arts heel kort 5 seconde aandacht was al teveel voor deze meneer	Behandeling	neg	pos	0.002395
Meteen doorgestuurd om foto's te laten maken	Communicatie	pos	neg	0.006567
de wegbewijssering	Zorg	pos	neg	0.008734
Dossier goed inlezen	Zorg	neg	pos	0.009868
Ik ben heel te vrede op de oogpollie ..	Zorg	neg	pos	0.011911
Personeel in dienst houden	Zorg	neg	pos	0.014603
De vervanger legde dat keurig uit	Zorg	pos	neg	0.016396
Was dik tevreden	Personeel	pos	neg	0.018509

Table 8.14: *H3 Dataset Top-10 Misclassified Responses with Lowest Confidence*

Text	***Aspect***	***True***	***Pred.***	***Logits***
attent personeel	Personeel	neg	pos	5.304
Persoonlijke aandacht door personeel	Zorg	neg	pos	5.226
Verpleging erg aardig en meelevend	Personeel	neg	pos	5.138
De ontvangst	Zorg	neg	pos	4.916
Netjes op tijd , vriendelijke en vaardige artsen	Vriendelijkheid	neg	pos	4.892
Goede zorg	Zorg	neg	pos	4.755
Snel en correct geholpen	Snelheid	neg	pos	4.752
De verpleegkundige en de artsen waren erg vriendelijk inlevend en mee denkend	Vriendelijkheid	neg	pos	4.741
De artsen waren ongelooflijk vriendelijk	Vriendelijkheid	neg	pos	4.737
Verpleegkundige waar zeer behulpzaam	Zorg	neg	pos	4.734

Table 8.15: *H1&2 Dataset Top-10 Misclassified Responses with Highest Confidence*

Text	***Aspect***	***True***	***Pred.***	***Logits***
Alles was goed , geen verbeteringen nodig	Behandeling	neg	pos	0.000389
Mijn poging om ook op	Zorg	neg	pos	0.001210
Bij het binnekomen	Behandeling	neg	pos	0.002491
Er is veel communicatie tussen de verschillenden afdelingen	Communicatie	pos	neg	0.002566
Aandacht is ook fijn en nodig	Zorg	neg	pos	0.002702
Ik heb deze opname ervaren als een warm bad	Behandeling	neg	pos	0.002834
1x nachtverpleegkundige	Wachttijd	neg	pos	0.003194
Een voorbeeld geven is moeilijk maar wat mij opviel is het feit dat de Eerste Hulpdienst de vergelijking met een geoliede machine goed kan doorstaan!!	Behandeling	pos	neg	0.003376
Fysio , had ik het gevoel dat het wel iets hielp	Behandeling	pos	neg	0.004051
Ik ben goed behandeld dus voor mij is alles goed	Zorg	neg	pos	0.004472

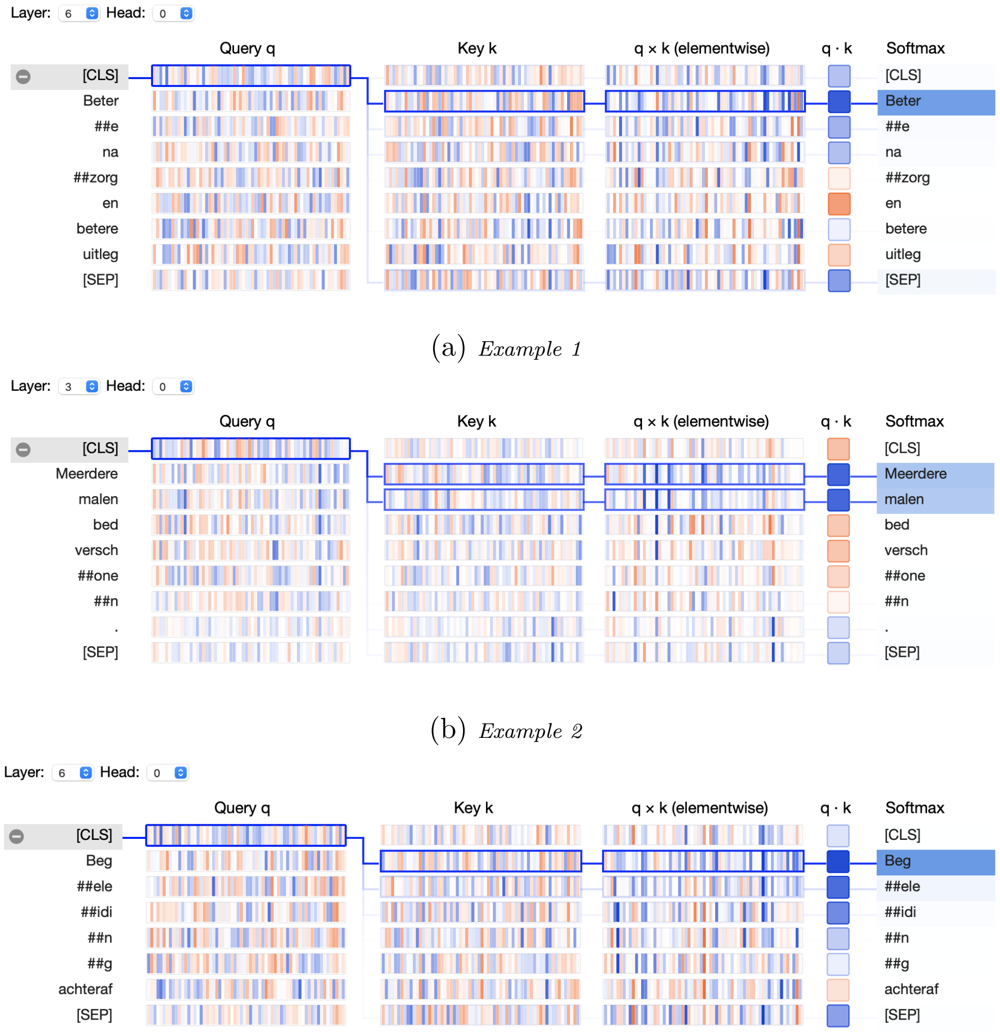
Table 8.16: *H1&2 Dataset Top-10 Misclassified Responses with Lowest Confidence*

	Text	***Aspect***	***True***	***Pred.***
1	Betere nazorg en *betere uitleg*	COMMUNICATIE	pos	neg
2	Meerdere malen *bed verschonen*	BEHANDELING	pos	neg
3	*Begeleiding achteraf*	ZORG	neg	pos
4	*behandeling en verzorging*	ZORG	neg	pos
5	*Persoonlijke verzorging*	ZORG	neg	pos

Table 8.17: The misclassified examples with high confidence and correct ground truth label. Italic parts indicate the aspect sequences

	Text	***Aspect***	***True***	***Pred.***
1	Dus tot onze *grote tevredenheid* : goed .	ZORG	pos	neg
2	*Uitleg* tijdens behandelingen is vaak aandachtspunt .	COMMUNICATIE	neg	pos
3	Alleen had ik problemen met *de taxi* .	COMMUNICATIE	neg	pos
4	En de wachttijd was *kort* .	WACHTTIJD	pos	neg

Table 8.18: The misclassified examples with low confidence and correct ground truth label. Italic parts indicate the aspect sequences



Layer: 6 Head: 0

(a) *Example 1*

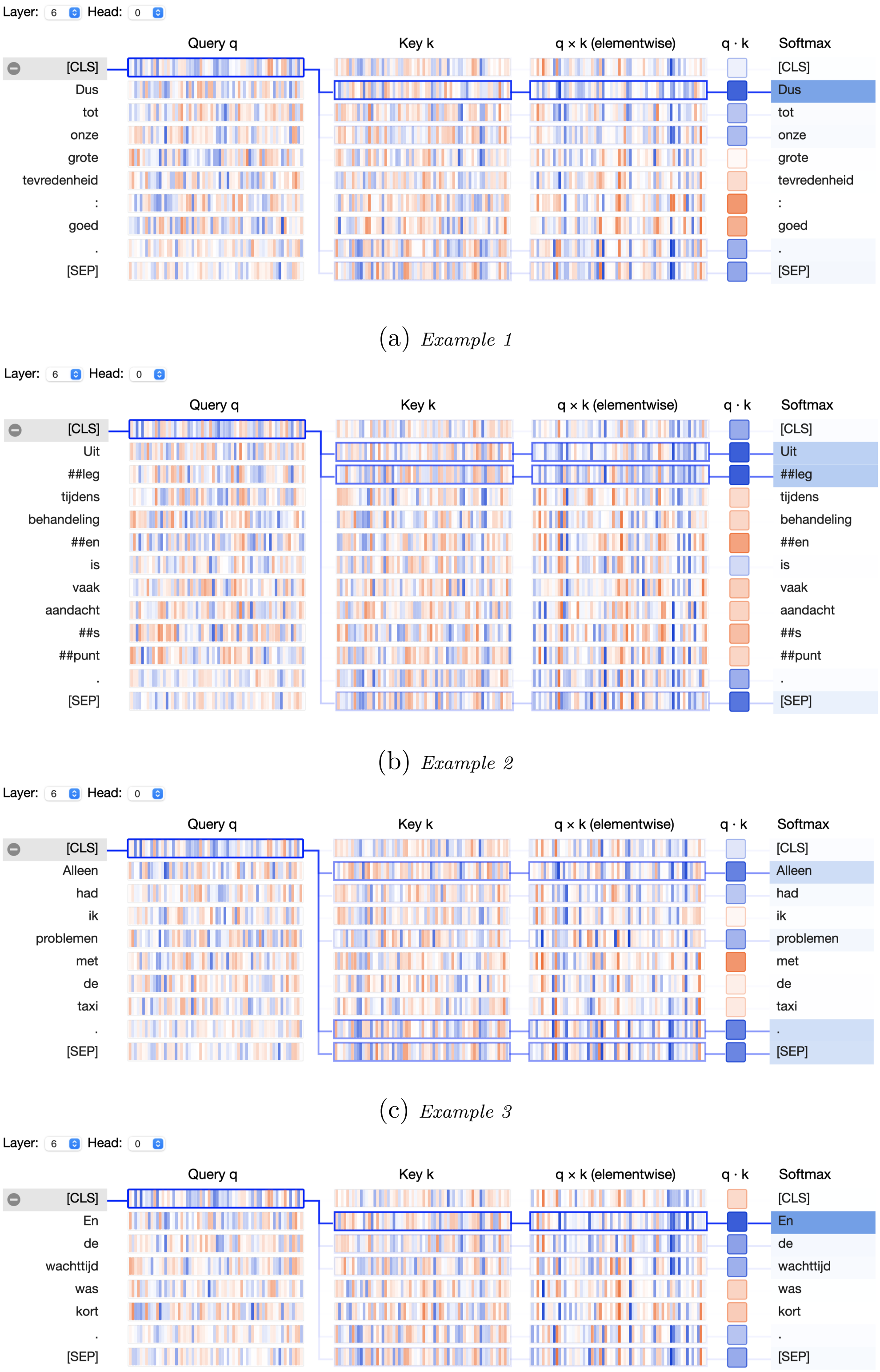
Layer: 3 Head: 0

(b) *Example 2*

Layer: 6 Head: 0

(c) *Example 3*

Figure 8.7: *Attention Weights Visualizations of Table 8.17, Blue denotes to positive and Orange indicates negative values*



Layer: 6 Head: 0

(a) *Example 1*

Layer: 6 Head: 0

(b) *Example 2*

Layer: 6 Head: 0

(c) *Example 3*

Layer: 6 Head: 0

(d) *Example 4*

Figure 8.8: *Attention Weights Visualizations of Table 8.18, Blue denotes to positive and Orange indicates negative values*

E Annotation Guidelines for Positive Aspects

Aspect	***Words***
Vriendelijkheid	vriend, vriendelijk
Communication	uitleg, voorlicht, duidelijk, informatie
Snelheid	snel, direct, fast, gauw, gezwind, haastig, spoedig, vlug, vlot
Zorg	ontvang, opname, behandel, zorg
Personeel	mens, arts, zuster, chirurg, verpleegster

Table 8.19: The words used to create the candidate pool for each positive aspect

Selected aspects
Vriendelijkheid
Communication
Snelheid
Zorg
Personeel

Table 8.20: *Pre-selected Positive Aspects*

These aspects are selected due to the frequencies of occurrences in the data. Selected aspects seem to be overlapping with each other especially with the aspect *Personeel*.

For example;

No	***Aspect***	***Sentence***
1	Zorg	ik ben heel erg tevreden over *de begeleiding* van de artsen ze luisteren goed en je hebt een eigen inbreng en ga in principe altijd vrolijk naar huis.
2	Snelheid	**het snelle bezoek** van de arts op zaal.
3	Zorg	**De kennis en behandeling** van de artsen en het verpleegkundig personeel.
4	Uitleg, Zorg	**Voorlichting** en **ontslag** doos zaalarts.
5	Zorg, Personeel	Bij het wakker worden op de IC zeer **goed opgevangen** door een verpleegster die zeer **bedreven** is in haar beroep.
6	Personeel, Personeel	**kundige** oogarts met **veel geduld**

Table 8.21: *General Examples*

This type of overlapping with personeel treated as one aspect other than personeel. This means first example belongs to the aspect ”Zorg”. Second example has the aspect “Snelheid”. Third example also has the aspect ”Zorg”.

Vriendelijkheid

This aspect includes the kindness, friendliness of personeel. This aspect strictly includes the word vriendelijk in the sentence.

Communicatie

This aspect includes communication-related word groups such as explanations, given information, speech, etc.

No	***Sentence***
1	De verpleegsters waren erg *vriendelijk*.
2	operatie is gelukkig meegevallen arts en opreatie kamer personeel waren *vriendelijk*
3	Bijzonder *vriendelijke* artsen en verpleegkundigen

Table 8.22: *Examples of Vriendelijkheid Aspect*

No	***Sentence***
1	*Informatie* die de arts gaf.
2	Het gesprek met de KNO-arts en zijn assistent was heel prettig en *informatief*.
3	*Communicatie* met chirurg
4	*Uitleg* over nieuwe medicijn

Table 8.23: *Examples of Communicatie Aspect*

Snelheid

If something is done quickly then it belongs to this aspect.

No	***Sentence***
1	Dat ik *direct* geholpen werd
2	*Snel en helder* opname
3	Dat ik *vrij snel* op zondag geholpen ben (geopereerd)
4	dat mijn dochter *heel snel* opknapte

Table 8.24: *Examples of Snelheid Aspect*

Zorg

This aspect has very broad context. Attention treated as in the context of care.

Personeel

If the response indicate satisfaction for the personeel but not in the preselected aspects then it belongs to this category.

No	***Sentence***
1	Onze longarts had heel *goed aandacht* voor mijn problemen
2	*De betrokkenheid, aandacht* voor de vragen die werden gesteld, zowel verpleegkundigen, artsen en van de service medewerkers.
3	*Goede controlle* van de artsen
4	Ik werd zeer *goed opgevangen* door de verpleging
5	volgens de arts *de ingreep*
6	*De begeleiding* van zowel verplegend personeel als artsen.
7	*Het maken van het audiogram* sloot goed aan op het bezoek aan de kno arts

Table 8.25: *Examples of Zorg Aspect*

No	***Sentence***
1	*Lieve* verpleegkundige en artsen
2	de zuster op de poli was *rustig en lief*
3	De artsen zijn heel *erg goed*, daar ben ik echt super tevreden over.

Table 8.26: *Examples of Personeel Aspect*

F Annotation Guidelines for Negative Aspects

Aspect	***Words***
Communicatie	communic, gesprek, inform, luister
Wachttijd	wacht, tijd, verkort, lang, verminder
Eten	eten, maaltijd, ontbijt, warm, drink
Behandeling	behandeling, infuus, operatie, ingreep, prik
Schoonmaken	schoon, kamer, toilet, vies, smerig, vuil

Table 8.27: The words used to create the candidate pool for each negative aspect

Selected aspects
Communicatie
Wachttijd
Eten
Behandeling
Schoonmaken

Table 8.28: *Pre-selected Negative Aspects*

Table 8.28 shows pre-selected negative aspects.
General Guidelines:

- Aspects may be consists of multiple words, however try to keep as small as possible (Example 1 and 2);
- Try to select the most important words for aspects (Example 3)
- In case of splitted aspects, select the most indicative part to annotate. (Example 4 and 5) In example 4; the word "lang" is more important indicative then "wachten", that's why the annotation should include the word "lang"

Examples:

No	***Aspect***	***Good***	***Bad***
1	Communicatie	*Betere informatie* over de napijn	*Betere informatie over de napijn*
2	Wachttijd	Erg vroeg aanwezig moeten zijn , dus *lang moeten wachten* op operatie	Erg vroeg aanwezig moeten zijn , dus *lang moeten wachten op operatie*
3	Wacthtijd	*lang moeten wachten*	lang moeten *wachten*
4	Communicatie	de informatie over de zelfmedicatie thuis na het ontslag is erg *onduidelijk*	*de informatie* over de zelfmedicatie thuis na het ontslag is erg *onduidelijk*
5	Wachttijd	*Lang* op de arts moeten wachten	*Lang op de arts moeten wachten*

Table 8.29: *Examples*

Communicatie

This aspect includes communication problems. Lacking of information should be treated in this aspect. Because the information could be given but did not.

No	***Sentence***
1	Graag zou ik wat *meer informatie* willen krijgen tijdens de behandeling over hoe het gaat.
2	Personeel *luistert niet* naar patiënten en *beantwoord vragen niet*
3	*communicatie* over de lange wachttijd voordat ik van de verkoeverkamer naar de operatiekamer ging

Table 8.30: *Examples of Communicatie Aspect*

Wachttijd

This aspect indicates the complaints about waiting times. If there are non-consecutive multiple words, we annotated the part we thought was the most critical.

No	***Sentence***
1	*lang wachten* op kamer
2	Wachttijd tussen opnametijd en gang naar operatiecomplex was *zeer lang* (meer dan 4 uur).

Table 8.31: *Examples of Wachttijd Aspect*

Eten

For this aspect, we annotated mainly the words that indicate the meal rather than the adjectives to describe them.

No	***Sentence***
1	*De maaltijden* kunnen beter
2	*maaltijden* zijn niet zo lekker en niet zo goed klaargemaakt ... gaar enzo
3	Smaak van *de warme maaltijd* niet gaar en zoutarm

Table 8.32: *Examples of Eten Aspect*

Behandeling

This aspect is related to the problems/complaints or complications during the treatment. Since there are many different words to describe a problem in the treatment process, deciding which words to annotate were hard. Again, we annotated the words we think are the most representative.

Schoonmaken

This aspect includes the adjectives which describe the cleaning problem.

No	***Sentence***
1	Dat er, tijdens het toebrengen van de pijnstiller *hersenvocht is afgenomen* wat niet de bedoeling was.
2	*Voor berijden* voor de operatie kamer
3	Verkeerd berekend in hoeveel tijd *het infuus* moest inlopen
4	*Zorg* aan patient direct na de operatie

Table 8.33: *Examples of Behandeling Aspect*

No	***Sentence***
1	*hygiene* van het toilet bij de kamer
2	De toiletten ter hoogte van bloedafname waren erg *smerig*
3	kamer *nooit schoon* gemaakt

Table 8.34: *Examples of Schoonmaken Aspect*