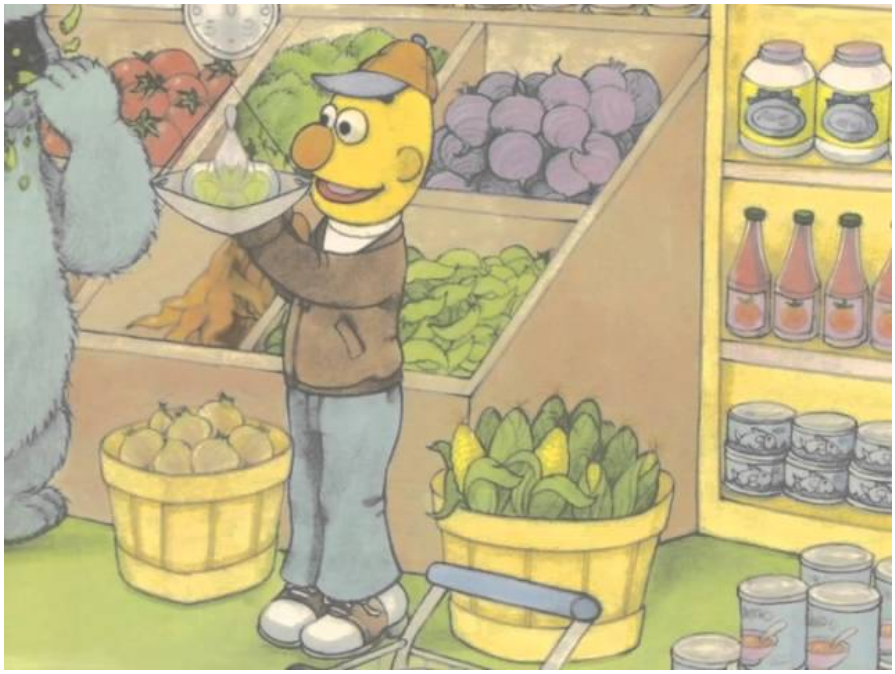
Applying Dutch pre-trained word embedding models to classify grocery products

submitted in partial fulfillment for the degree of master of science

Gijs Gubbels
11408707

master information studies
data science
faculty of science
university of amsterdam

2022-07-06



	Internal Supervisor	External Supervisor
Title, Name	dr. Reshmi Gopalakrishna Pillai	Marjolein Christiaanse
Affiliation	UvA, Information Studies	Albert Heijn, Product Data
Email	r.gopalakrishnapillai@uva.nl	marjolein.christiaanse@ah.nl

Universiteit van Amsterdam

Amsterdam Data Science

Albert Heijn

Applying Dutch pre-trained word embedding models to classify grocery products

Gijs Gubbels
University of Amsterdam
Amsterdam, The Netherlands
gijs.gubbels@student.uva.nl

ABSTRACT

Correct product categorization is important for retailers in order to help consumers navigate through their assortment. The automatic classification of products can reduce work time and errors. A number of studies have already shown that BERT can play a prominent role in this task. However, these models are mainly pre-trained and fine-tuned on English product data collected at a later stage in the product data flow. In this study, we investigate whether a fine-tuned BERTje model can be used for the product classification task based on Dutch supplier data. This supplier data consists mainly of descriptions that have not been checked on their quality. BERTje has been compared with TF-IDF and the Dutch pre-trained Word2vec and fastText in combination with an XGBoost classifier. The TF-IDF experiment proved to be a solid baseline, scoring an MCC score of 82.21% for the categories in the bottom layer of the taxonomy. Fine-tuned BERTje achieved the best results with an MCC score of 95.40% on predicting categories in the bottom layer of the taxonomy. This study showed that BERTje is better at predicting products based on short and concise supplier descriptions than other pre-trained Dutch models.

KEYWORDS

retail, product classification, NLP, BERTje, supplier data, word representation, fastText, Word2vec, pre-trained models

1 INTRODUCTION

1.1 Product data is everything

The popularity of e-commerce has been growing rapidly over the last decade. During the COVID-19 period, this e-commerce trend has accelerated [2]. Especially, the click-and-collect and grocery delivery options are used more than ever. The online grocery sector will continue to grow and might account for 18 to 30% of the food-at-home market in leading countries by 2030 [27]. The evolving customer behavior and technology development are the two drivers of the rise in popularity. For instance, the features of retail websites and applications such as recommendations, product taxonomy and product comparisons are considered valuable by consumers [6]. All these features contribute to users searching for the products they are looking for.

The hierarchical tree structure of a product taxonomy allows customers to easily browse through an enormous quantity of products by narrowing down to specific subcategories [32]. Manually assigning products to the corresponding categories in a taxonomy has a number of drawbacks. Product assortments could contain millions of items and vary on a regular basis. The ever-changing assortments make manually categorizing products a hard and time-consuming task. Secondly, manual categorization comes with the risk of human errors. Thirdly, assigning products to categories is a subjective task. Employees may assign a particular product to different categories because they perceive the category or product differently. These risks caused by human actions may negatively affect the performance of retail stores. Therefore, research in the retail sector is focusing on preventing these problems using machine learning (ML).

The key to successfully implementing machine learning is the quality of data on which models are trained and evaluated. The data assessment is an essential stage in the data flow. Retailers and brand manufacturers often have no agreement on a standard process to maintain data quality [12]. The flaw of supplier data is that it is lacking quality. This has a number of reasons: vendors often do not care if the data is one hundred percent correct, they are sometimes bound by a maximum number of characters, and different suppliers have different ways of describing their products.

1.2 NLP to the rescue

Product categorization, also known as product classification, is the task of automatically predicting the taxonomy path for a product. Due to the number of products and categories, the imbalance of categories and noisy product data, the task remains to be a complex challenge. When a product taxonomy contains numerous levels and categories, the task becomes more complex because the corpus gets bigger and the similarity of categories may be close to one another. Therefore, the need to capture the meaning of words and phrases as accurately as possible is crucial nowadays. This subject has been extensively studied in the past few years. Due to, in particular, the developments of natural language processing (NLP), current literature is focusing on how (pre-trained) models such as Word2vec [21], fastText [5], GloVe [24] and BERT [10] can be used to approach the product classification task based on unstructured textual product data.

Studies are primarily concerned with product titles and descriptions of high quality. This data is generated and assessed by the retail company itself based on data provided by suppliers. However, automatic product classification can be of greater value when it is used at the stage where suppliers are providing the product data. By doing so, suppliers do not have to fill in unnecessary attributes and will therefore be able to process the product data faster. For example, when a supplier submits a coffee product, it is not interesting to ask the supplier to also fill in the alcohol percentage attribute, but rather whether the product contains caffeine. Thus, categorizing the product as a coffee product at an early stage can make the process more efficient.

The majority of the studies focus on English product titles and descriptions. However, there are still many challenges regarding product data in other languages. Other languages are semantically different than English and there are differences in sentence structures.

1.3 Focus of the study

The focus of this paper is to measure the performance of BERT pre-trained on the Dutch language. The research question this study aims to answer is:

- To what extent does a fine-tuned BERTje model improve the classification of products into a product taxonomy based on unstructured textual attributes when compared to TF-IDF, Word2vec and fastText?

The structure of the study is supported by one sub-question:

- How do pre-trained word embedding models such as Word2vec and fastText represent product descriptions in the classification task when compared to a not pre-trained word embedding model such as TF-IDF?

A variety of word embedding techniques will be studied and in particular the performance of a fine-tuned Dutch pre-trained model BERTje. The contributions of this research are listed below:

- We compare four different word representation techniques with regard to the product classification task. Both conventional and advanced embedding techniques are studied to serve as strong baselines.
- The Dutch pre-trained BERTje model has been fine-tuned on a product classification task with promising results. Depending on the predefined product taxonomy, the model can be used as a solid foundation for other product classification tasks.
- The model will be trained and evaluated on supplier data in an early stage of the product data flow. This can make the data flow more efficient.

As this research is done in the form of an internship at Albert Heijn, the data used contains product data of the assortment of Albert Heijn. Section 2 will elaborate on the literature background in the field of product classification. Then, the experiments will be explained in section 3. In section 4, the results will be presented. After which the discussion (section 5) and conclusion (section 6) will follow.

2 RELATED WORK

Automatic classification of products into a hierarchical taxonomy is a task that has received attention from a number of studies [19, 22, 34]. The complexity and size of the data cause researchers to explore different approaches. The emergence of new NLP techniques such as transformers and transfer learning, has made it possible for studies to successfully accomplish this complex task. A brief summary of the most important techniques and studies in the area of product classification will follow.

2.1 Product attributes

Attributes are characteristics of a product and thus are valuable for the input of classification models. Examples of attributes are descriptions, titles, price and, image. A frequently used type of product attribute is the textual attribute [22, 34]. Product descriptions can contain details of the product: *Coca-cola zero can 0.15l 1x, soft drink with plant extracts, with sweeteners.* In contrast, product titles are usually brief and concise so that customers immediately know which product they are dealing with: *Coca-cola zero can.* It makes it harder for classifiers to classify products correctly due to the limited word count and thus lack of information [16, 22]. Several studies took both the product descriptions and titles as input for their classifiers [34, 37].

The impact of certain pre-process stages, such as removing stop words or stemming, on the performance of a product categorization model has also been studied [36]. The study concluded that stemming and stop word removal do not have to be applied to product title classification since product titles are very brief. Bigrams can be useful for titles and short descriptions because of their small length.

With the rise of deep learning, multimodal learning techniques have also been explored where different types of unstructured product attributes (text and images) are combined to predict the correct categories [3, 8, 32]. In addition, researchers analyzed the effect of the combination of structured (price and user clicks on the website) and unstructured (description and titles) attributes on the performance of hierarchical classifiers [19].

After reviewing the literature, we noticed that the techniques used and the conclusions drawn do not apply to every situation. The product data can vary widely from one study to another in terms of cleanliness, length, and structure. The mentioned studies used assessed data from (online) retail stores and not directly from suppliers. Secondly, to the best of our knowledge, none of the studies have used Dutch data which is semantically different than English.

2.2 Text representation

An important part of classifying products is transforming textual data into usable vectors. Term-frequency vectors such as Bag-of-Words (BOW) and TF-IDF have been successfully applied to product classification problems [13]. An important advantage of these methods is their simplicity. These simple techniques in combination with classifiers can potentially outperform more novel NLP techniques [35]. However, the simplicity comes with a number of drawbacks: 1) the location of words in sentences is rarely taken into account, 2) the production of high-dimensional dense matrices, 3) the semantic meaning of words and the connections to other words is ignored.

More advanced word embedding models such as Word2vec [21], fastText [5] and GloVe [24] have been developed to capture the semantics of words as accurate as possible. Research has shown that all three models combined with a classifier are suitable for a text classification task with recall scores ranging from 87% to 90% [31]. Also within the area of product classification, the word embedding models have proven to be useful [23, 35]. One challenge is that these models require a large corpus to have enough training data. In the case of a small training corpus, transfer learning with the use of pre-trained word embedding models provides a work around. Another challenge is that these static word embeddings

cannot properly handle words that can have multiple meanings: polysemous words. Each word is presented by only one vector.

More recent these drawbacks were solved with the introduction of contextualized word representation models, such as BERT [10] and ELMo [25]. The former is used in several product classification researches and outperforms the more traditional word embedding techniques [22]. A groundbreaking innovation in the NLP research field is the transformer architecture. In the current NLP literature, the pre-trained model: Bidirectional Encoder Representation with Transformers (BERT) has proven to be a state-of-the-art language model. BERT is distinguished from other transformer models by its capability to learn information in a sentence from left to right and right to left. The model is trained with masked language modeling, which masks off a proportion of a sentence and then predicts the masked words. This method enables it to assimilate the location per word in a sequence and includes it in the embedding. Research showed that bidirectionally trained language models perform better than single-direction models in understanding the context of language [10]. Depending on the task, BERT can be fine-tuned by post-training the model on a specific labeled data set [29]. This is how BERT also can be used for text classification. Several studies have investigated the performance of language models like BERT to assign products to the correct category [4, 16, 22, 26, 37]. Although BERT achieves new state-of-the-art performances, there are still challenges regarding different aspects of the data such as the language of the structure of product descriptions. Because BERT is trained on multiple languages, BERT models have been developed for specific languages to be able to deal with foreign languages. For example, BERTje trained on Dutch language outperforms the multilingual BERT model on tasks like sentiment analysis and part-of-speech tagging [9].

2.3 Hierarchical and flat classification

Within the current literature, a difference can be made between studies that classify text into hierarchically structured categories [16, 22, 34, 37] or flat constructed categories [7, 15]. Flat classification considers only the leaf nodes (subcategories) as the predefined labels where no relation between these categories is considered. Flat models have been compared to hierarchical models [19]. In the research the flat classification approach performed better than the hierarchical method, which seems counterintuitive because the flat approach ignores the hierarchically structured information.

In contrast, hierarchical classification takes the structure of the taxonomy into account. Two kinds of hierarchical approaches have been studied in the current literature: 1) a global approach considers the whole taxonomy using just one more complex classification model [28]; 2) a local approach uses a classifier for each parent node [17]. [34] conducted research on a BERT-based ensemble model. Instead of looking at the product hierarchy from a flat perspective, the researchers focused on the hierarchy level by level. According to the researchers, this has an advantage because in this way the child nodes are cut off from the current parent node. Unrelated information is thus excluded at the specific parent node.

Another more creative approach is using machine translation as a solution to product categorization [20, 30]. Instead of looking at the problem as a standard classification task, the product description is translated into a root-to-leaf path. Advantages of this technique is its capability of ignoring ambiguity and noise in texts and its ability to output non-existent paths between nodes in a taxonomy.

Since the word embedding techniques are the primary focus of this study, flat classification was chosen. Additionally, the taxonomy used is rather small (3 layers and 29 leaf nodes) compared to previous studies, therefore hierarchical classification is not essential. Last but not least, we have observed that we also gain insights about the current taxonomy from the errors made by the classifier.

3 METHODOLOGY

This research attempts to create a setup for a classification model that categorizes Dutch products into a predefined taxonomy. By fine-tuning the Dutch pre-trained BERTje and comparing its performance with different baseline models, the aim is to find out whether BERTje is useful for classifying products based on Dutch descriptions. The method of the research consists of a number of elements: 1) Gathering the data 2) Pre-processing and exploring the data 3) Creating product description representations 4) Training classification models 5) Evaluating the performance of the models. All code used in this research can be found in the Github repository[1].

3.1 Data

The product data of Albert Heijn describes 46.570 unique products. To train and evaluate the performance of the models a solid labelled data set is crucial. However, the quality and completeness of the labels of the total product data set is lacking. Therefore, together with product data specialists of Albert Heijn a scope of the product taxonomy has been defined for this research. A total of 6.808 correctly labelled products remained for training and evaluation. The scope can be expanded later if necessary, on which the model can be re-trained.

3.1.1 Supplier data. A distinction is made between the data provided by the supplier of the products and the data generated and assessed by data specialists of Albert Heijn, since these are different stages in the product data flow. The research focused on the data that suppliers provide and not the data that Albert Heijn generates at a later stage. In this way, a category can be assigned to a product as early as possible in the product data flow so that the supplier only is required to fill in relevant attributes for a particular product.

The product data is composed according to the GS1 guidelines [1]. Each product has a unique identifier called *gtin* and each supplier has a unique identifier called *gln*. Since the research focuses on various NLP techniques only textual description data is incorporated. The ingredient list was left out of the scope of this study because it provides less information about a product's place in the product taxonomy than a description. Also, the brand name was left out of the scope since this can cause confusion for the classifier [22]. The different textual fields and their definitions are listed below:

- **Functional name** - The purpose of the functional product name is to describe the product as unique as possible. This field must be filled in as richly and specifically as possible by the supplier and has a maximum limit of 35 characters.

[1] https://github.com/Gauss123/product-classification

Textual field	Example
Functional name	ice tea green
Trade item description	lit green pet 500ml x 12
Label description	lipton green ice tea 500 ml
Regulated product name	koolzuurvrije frisdrank met groene thee-extract met suiker en zoetstof

Table 1: Example of supplier data

- **Trade item description** - The purpose of the information in this field is to immediately recognize a product and distinguish it from other products. This field contains the unique description of a product including product and packaging information and has a limit of 200 characters.
- **Label description** - This field contains a description that is clear enough for consumers to recognize and identify this item when they purchase it on a website. However, not all text on the label needs to be entered. The maximum length is 500 characters.
- **Regulated product name** - This is the legal name of the item. Additions such as added sweeteners, sugars and prepared with fruit are part of the legal name. The maximum length of this field is 500 characters

The supplier determines which texts are used to describe the products. Examples of the descriptions of a beverage product are shown in table 1.

3.1.2 Product taxonomy. The product taxonomy is composed of three levels, with 3 distinct categories on the first level, 7 categories on the second level, and 29 categories on the third level. The English translations of all categories can be found in Appendix A. Each product is assigned to one category at each level in the taxonomy, making the classification task single labelled. Figure 1 shows the distribution of the third-level category labels of the labelled dataset. The majority of products are allocated in smaller categories (categories with less than 6% of the labelled dataset). It is also noticed that the majority of the products are beverage products. This imbalance adds to the complexity of the classification task. The exact number of products per category is provided in Appendix B.

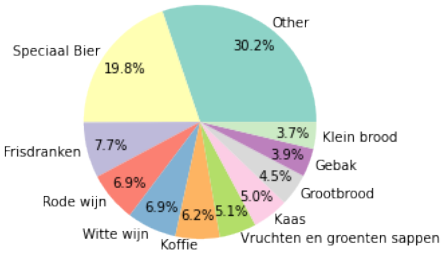


Figure 1: Distribution of the categories on the third level

3.2 Pre-processing and Exploratory Data Analysis

In order for the classifiers to perform optimally, pre-processing of the textual data is required. The steps in the pre-process pipeline are discussed below. Certain steps were skipped depending on the word embedding technique. Since the tokenizer function of BERTje has the ability to take sentences as input that are not cleaned, the pre-processing steps were not followed for the input data for this model. In section 3.4 we will discuss how the different descriptions were used as input for the model. During the pre-processing, we performed exploratory data analysis (EDA) to support certain pre-process steps and get a better understanding of the data.

(1) **Lower casing descriptions** - All textual description data was transformed to lowercase text. This ensured that the same words, regardless of lowercase or uppercase, were treated as equal in later stages.
(2) **Removing numerical characters** - Product descriptions can include the weight or ingredient information which are expressed in numbers (for example: *'teisseire mojito 0% 60cl'*). We decided to remove these numerical characters (0-9) as this study focused on textual attributes.
(3) **Handling punctuation** - Descriptions may have some structure with respect to punctuation. A punctuation character can be a connection between two words (for example: *'zwarte theeën met kruiden en/of cacao en/of fruit en/of natuurlijk karamel aroma'*). Therefore the choice was made to replace these characters with a space: '/', '\', ':', and '-'. The other punctuation characters were removed.
(4) **Remove excessive spacing** - Choosing to replace certain punctuation with spaces can lead to excessive white space. For this reason, the excessive spaces were removed.
(5) **Removing Dutch stopwords and other irrelevant words** - A set of commonly used words in Dutch was removed from the descriptions. The Dutch stop words list of NLTK[2] was used for stop word removal. In addition, words consisting of 1 letter were also eliminated.
(6) **Lemmatization** - In order to normalize the textual data, we used the Dutch spaCy lemmatizer. This lemmatizer was chosen because it achieved promising performance according to its developers[3].

3.2.1 Missing data and short descriptions. Next, the missing data and length of the descriptions have been examined. Due to the significant amount of missing descriptions, we decided to merge the different descriptions (table 2). The different descriptions are grouped together with a space in between. No particular order was taken into account; the order of the columns in the Albert Heijn data set was maintained. Once the descriptions were merged, one product was left with an empty description. This product is excluded from further research.

Another problem is the small length of descriptions. Having few words results in less information about the products, which might be harmful to the performance of the classifier. The merging of descriptions (M = 8.96, SD = 3.75) also deals with this problem

[2]https://github.com/xiamx/node-nltk-stopwords/blob/master/data/stopwords/dutch
[3]https://spacy.io/models/nl

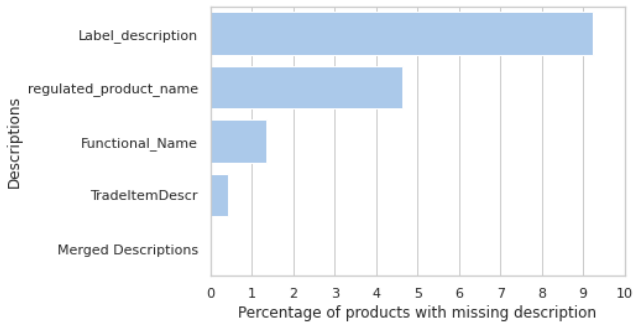


Figure 2: Percentage of missing values per description (including merged description

of short descriptions as can be seen in the box plot figure (figure 3). The interquartile range is higher than the other descriptions. Therefore, for further data analysis, the focus will be on the merged descriptions.

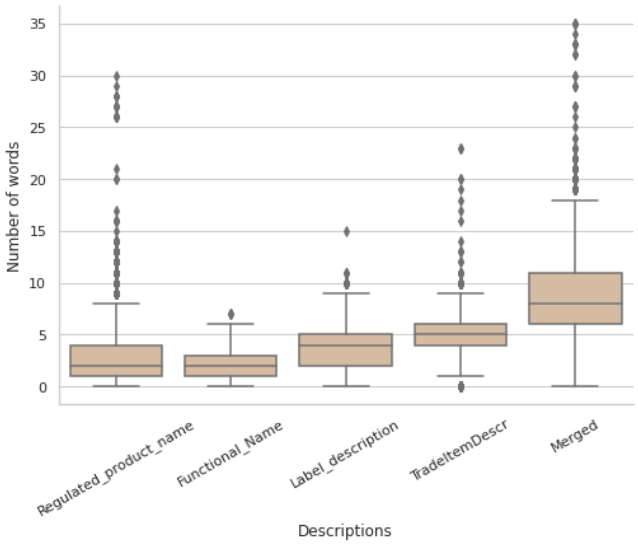


Figure 3: Boxplots of word count per description

A word cloud for each main category was created to analyze the most common terms. The size of the words in the word clouds indicates how often a word occurs in the corpus for each main category (Appendix C).

3.3 Baseline

A number of baseline experiments were created to gain insights from different word embedding techniques and compare them with BERTje.

3.3.1 Word embeddings. Following the pre-process pipeline, the textual data is vectorized for later use as input for classification models. This transformation has been done through three different word embedding models. The TF-IDF score and the Dutch pre-trained Word2vec [11] and FastText [14] are used to create word representations to compare with the word representations of BERTje. Each word embedding model has its own advantages and disadvantages.

TF-IDF is a basic statistical measure to give every word in the corpus a score. Depending on the data and the task, it can be a simple way to recognize the most descriptive words from a text. However, it lacks the ability to recognize context of words in a text. Word2vec is a more advanced technique. The two-layered neural network seeks an understanding of the context of words by looking at the neighboring words. The main disadvantage is that Word2vec captures a small context and does not look at the global context of words. On the contrary, fastText's advantage over Word2vec is that it uses subword information to return vectors for OOV (out-of-vocabulary) words. The choice to use pre-trained word representations over training them from scratch, was due to the fact that the descriptions are short and concise which results in a small train corpus. The 5 most similar words for 'bread' are shown in table 2 for the two pre-trained word embeddings.

#	FastText	Word2vec
1	'brood.'	'vlees'
2	'stokbrood'	'roggebrood'
3	'melkbrood'	'koeken'
4	'broods'	'gebak'
5	'broodjes'	'broodjes'

Table 2: Top 5 most similar words for 'brood' (bread)

Padding was used for all three word embedding techniques (FastText, Word2vec and BERTje) to create descriptions of the same length. The tokenized descriptions are padded with zeros to a length of 35, since this is the maximum length of the descriptions.

3.3.2 Classification models. As a baseline model, we used one classifier, the XGBoost classifier, to compare the outcomes of the baseline experiments. For the product classification task, XGBoost proved to be a solid baseline classifier in combination with Word2vec and fastText with F1 scores of 80.3% for Word2vec and 86.8% for fastText for predicting main categories [22]. The baseline model is relatively simple and has a probability of achieving adequate performance.

3.4 BERT(je)

Several studies have shown that BERT is a new state-of-the-art model to deal with the product classification task. However, the studied BERT variants are pre-trained on English texts and take English product descriptions as input. Because the Dutch language differs in semantics and the structure of sentences is different in both languages, it is important to study the monolingual pre-trained BERT: BERTje. In this section, we will give a brief overview of the functioning of BERT in a classification task.

Figure 4 displays the overview of the tokenization and classification process. It should be noted that no pre-processing of the descriptions has taken place. The four different descriptions (label description, regulated product name, functional name and trade item description) are treated as four distinct sentences.

BERT uses WordPiece to tokenize the words by splitting them into word pieces if necessary (e.g. 'snowing' becomes ['snow',

'##ing']) [33]. Also, with the help of special tokens, the data has to be transformed to a required format for BERT to understand the input. The four sentences are separated with the separation token [SEP]. The beginning of the merged product description is indicated with the [CLS] token. This token is used for classification tasks. Due to the varying length of the descriptions, the descriptions are padded with the [PAD] tokens to equal the lengths. Instead of words BERT represents tokens, each representing a (sub)word. The token IDs of the vocabulary of BERT, which it learned during pre-training, are assigned to the corresponding words. By using an attention mask BERT will focus on the non-padded tokens and ignore the padded ones.

BERT consists of 12 similar layers stacked on top of each other. Each layer takes a list of word representations as input of the last layer and generates the same amount of word embeddings as output. The hidden state corresponding to the [CLS] token is used in the classifier. BERT is fine-tuned for the classification task by training it on a labelled dataset. The BertForSequenceClassification is used to add a single linear layer on top of the pooled output to classify the product descriptions into a category[4].

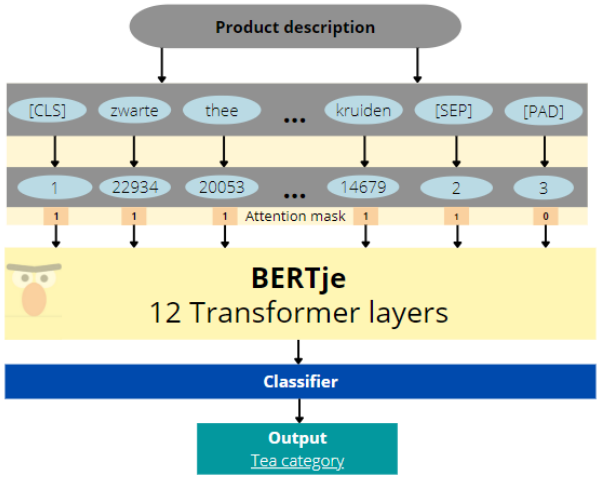


Figure 4: Product classification with BERTje

3.5 Evaluation metrics

A single evaluation metric cannot reflect the benefits and drawbacks of the classifiers en word embedding techniques [18]. Therefore following the metrics proposed by this review the accuracy, and weighted average precision and F1 score are used as the evaluation metrics. The accuracy is a simple metric because it measures the percentage of correctly predicted categories. A weighted variant of the metrics was chosen because the score takes into account the imbalance of categories. The support of each class is taken into account by the formula. The F1 score is the harmonic mean of the precision and recall. It takes the false positives and false negatives into account, which is important in our case with regard to the class imbalance.

However, these metrics can still misrepresent imbalanced classes. Therefore, the Matthews correlation coefficient (MCC) was also included in this study. MCC is a score that is a more balanced statistical measure. All fields in the confusion matrix (true positives, true negatives, false positives and false negatives) are included in the calculation in proportion to the number of negative and positive classes in the dataset:

$$\text{MCC} = \frac{\text{TP} \times \text{TN} - \text{FP} \times \text{FN}}{\sqrt{(\text{TP}+\text{FP})(\text{TP}+\text{FN})(\text{TN}+\text{FP})(\text{TN}+\text{FN})}}$$

3.6 Experimental setup

This section will discuss the experimental setup of this research.

3.6.1 Train and evaluation set. The data is split into training and evaluation sets using stratified KFold[5]. This data splitting technique splits the data into 5 folds which allows the model to train and evaluate a wider variety of the data. Stratified sampling is chosen over the regular KFold strategy to ensure the percentage of each category will be similar in each fold and thus each fold accurately represents the original data.

3.6.2 Taxonomy levels. Each experiment was performed for each of the three different layers in the taxonomy. This enabled us to analyze the performance when the number of product categories increases.

3.6.3 Baseline models. To compare the performance of our BERTje model, TF-IDF and the Dutch pre-trained Fasttext and Word2vec embeddings were considered. It should be noted that the pre-trained models were trained using different methods and corpora. Table 3 shows the characteristics. The fastText model was trained using CBOW on Dutch texts from Common Crawl and Wikipedia sites. The pre-trained Word2vec model was trained using Continuous Skip-gram on the Dutch CoNLL17 corpus. Vectors with only zeros were selected to represent OOV words for the Word2vec experiment.

	FastText	Word2vec
Architecture	CBOW	Skip-gram
Vector size	100	300
Window	5	10
Vocabulary size	2.000.000	2.610.658

Table 3: Parameters of pre-trained word vectors

3.6.4 BERTje. BERTje is pre-trained on multiple Dutch corpora. The vocabulary represents 30.073 tokens and the dimension of the vectors is 768. More specifications can be found in the study [9]. The values for the parameters tested of BertForSequenceClassification were chosen based on the recommendations of the developers of BERTje [9] and are shown in table 4. As BERT has 110 million parameters, extra attention was given to prevent the model from

[4]https://huggingface.co/docs/transformers/model_doc/bert#transformers.BertForSequenceClassification

[5]https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html

overfitting. Therefore dropout regularization has been added and the training and validation loss was plotted for the epoch range (Appendix E). The validation loss decreases together with the training loss. At epoch 4 the loss seems to stagnate.

Parameter	Values	Optimal value
Batch size	16, 32	32
Epochs	2, 3, 4	4
Optimizer	Adam	Adam
Learning rate	5e-5, 3e-5, 2e-5	3e-5
Dropout	0.5	0.5

Table 4: Optimal parameters

4 RESULTS

This section will show the results of the different experiments.

4.1 Word representations

An important aspect of pre-trained word embedding models is dealing with OOV. After pre-processing the data, the Word2vec model recognized 49.45% of the unique words in the corpus. It was examined whether more words would be recognized if lemmatization was not applied. This resulted in the Word2vec recognizing 32.51% of the unique words. Since the model recognizes more words from the corpus, we decided not to apply a lemmatizer for the Word2vec and fastText embedding models.

The product description embeddings were visualized in a 3D plot for every word embedding technique with the legend being the categories of level 1 5. The plots for the other taxonomy levels are presented in Appendix D. Using t-distributed stochastic neighbor embedding (t-SNE)[6], the high dimensional vector representations were transformed to three-dimensional vectors. The algorithm tries to preserve only small pairwise distances or local similarities, thereby minimizing the information loss.

4.2 Classification results

After training, the performance of the different models was measured for each level of the product taxonomy. The classification results are presented in table 5. The model with the highest MCC score for all three levels in the taxonomy was the fine-tuned BERTje model. We observe that the classification model that used TF-IDF scores outperformed the pre-trained Word2vec and fastText, except for predicting level 1 categories where fastText has a slightly higher MCC score. Although the Word2vec experiment performs the poorest, it still has a level 3 MCC score of 75.19%.

5 DISCUSSION

In this section, the research questions are answered by analyzing the results in more detail. Subsequently, the limitations in terms of validity and generalisability of this study will be discussed.

[6]https://scikit-learn.org/stable/modules/generated/sklearn.manifold.TSNE.html

5.1 Sub-question

In this section the sub-research question is answered:

- How do pre-trained word embedding models such as Word2vec and fastText represent product descriptions in the classification task when compared to a not pre-trained word embedding model such as TF-IDF?

When visualizing the embeddings in a 3D scatter-plot, the TF-IDF embeddings seem to separate the categories on the first level better than the Word2vec and fastText embeddings (figure 5). Also when the second and third levels of the taxonomy are considered, TF-IDF seems to better represent the product descriptions (Appendix D). However, these visualizations were created after applying dimensionality reduction and we might have lost crucial information. The classification results confirm the assumption since TF-IDF outperforms the pre-trained word embeddings for all levels in the product taxonomy.

A remarkable result is that TF-IDF achieves an MCC score of 82.21% and an F1 score of 83.20% for categories on the third level of the taxonomy. In several studies, pre-trained word embedding models are used as a baseline, however, the advantage of TF-IDF is ignored in comparison with more advanced models such as GloVe, Word2vec and fastText [16, 22, 37]. When analyzing the differences between the levels in the taxonomy, the decrease in performance of the TF-IDF experiment is the smallest based on all metrics. The decrease in MCC score when shifting from level 1 to 3 is 5.65%, while Word2vec decreases by 10.93% and fastText decreases by 11.93%. This result is not consistent with previous comparisons made between TF-IDF and pre-trained embeddings in a classification task, where accuracy decreased as the number of categories increased [15]. However, the difference in result may be that the study combined TF-IDF scores with a Logistic Regression classifier, while our study used an XGBoost classifier.

These variations in representations could be related to the corpora that the pre-trained models were trained on. The pre-trained models do not recognize some words in the product descriptions simply because they have not been introduced to these terms during training. As stated in section 4.1, 32.51% of the unique words in the corpus are unknown to the Word2vec model. As a result, Word2vec may be unable to recognize significant (sub)words. The fastText model is less affected by this since it uses character n-grams to retrieve sub-word information. Words that are written together by suppliers and appeared during the pre-training of fastText, can therefore be split by the model. Examples of these kinds of terms are: *weightcheese*, *clbeer* and *sweetenercola*. Other sources of OOV words are brand names (e.g. Oatly, Nestle and Streeckgenoten) and misspellings ('aarbei'). Also, the used corpus in this research is rather domain-specific (retail products), while the word embedding models are both pre-trained on a wider variety of domains.

Despite the fact that Word2vec and fastText are seen as more advanced techniques than TF-IDF, the latter seems to represent the descriptions better when we consider the different levels of the taxonomy. Since TF-IDF is a word frequency statistic, it has the benefit of not requiring any pre-training. In this way, the technique determines the degree of similarity between the product descriptions. OOV terms that do not appear frequently in the categories

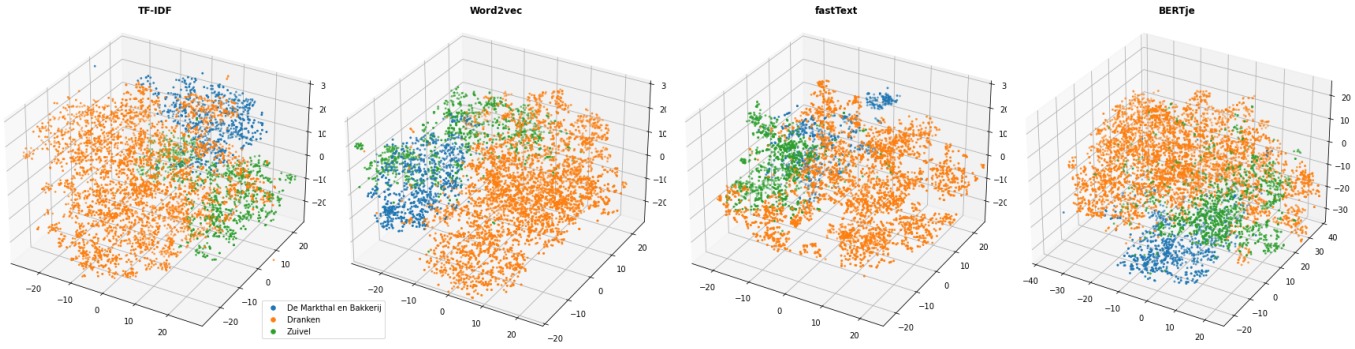


Figure 5: Product embeddings with level 1 categories as labels

	Accuracy			Precision			F1			MCC		
Taxonomy level	1	2	3	1	2	3	1	2	3	1	2	3
TF-IDF	94.20	91.51	83.52	94.14	92.23	84.74	94.10	91.38	83.20	87.86	89.98	82.21
Word2vec	92.87	88.25	77.09	93.20	88.56	77.17	92.90	87.98	75.94	85.58	86.04	75.19
fastText	94.14	88.88	77.86	94.28	89.55	77.96	94.11	88.58	76.77	87.95	86.85	76.02
BERTje	99.04	98.46	95.74	99.04	98.47	95.94	99.04	98.45	95.76	98.03	98.16	95.40

Table 5: Performances (%) of word embeddings for different taxonomy levels

will receive a low score, while words that appear more frequently will receive a higher score.

5.2 Research question

This section will answer the research question:

- To what extent does a fine-tuned BERTje model improve the classification of products into a product taxonomy based on unstructured textual attributes when compared to TF-IDF, Word2vec and fastText?

The fine-tuned BERTje model outperforms the other word embedding models on all metrics. The confusion matrices for the two word embedding models (Figures 8 and 7) show that the errors are more spread out in the TF-IDF experiment than in the BERTje experiment. As can be seen in the confusion matrix in figure 6 the majority of the mistakes are made with dairy products. When predicting the first level categories in the taxonomy, the most errors are made when the model confuses dairy products with beverages products. To us, however, this is not incorrect as drinking dairy could also be considered a beverage category. Also, the model confuses eating dairy products with bakery products and vice versa (figure 7). These mistakes are primarily due to products containing cream and butter.

This leads to a crucial aspect of a product taxonomy. The current product taxonomy is a single labelled dataset. However, the errors of the model state that categories have a certain degree of overlap which implies the need for products with multiple labels. Especially when the taxonomy expands with more products and categories, there will be a greater degree of resemblance between (sub)categories and products.

The results of the experiments are shown in table 5. All models perform better when targeting categories at a higher layer in the taxonomy. This is in line with the results of [22]. A category in a higher layer consists of more products, which means more training and evaluation data. Also, the number of classes in the upper level is lower than in the lower levels, making it easier for a classifier to classify products correctly.

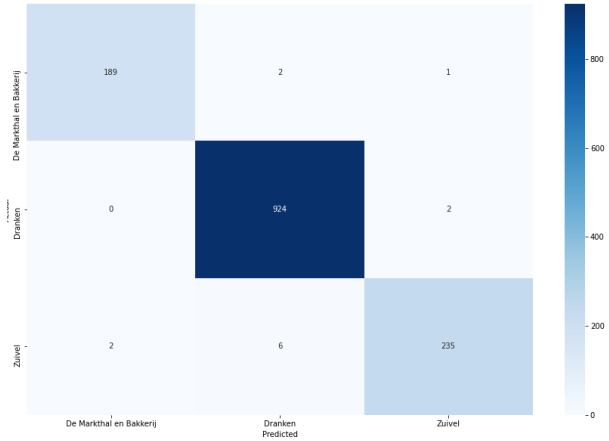


Figure 6: Confusion matrix of the BERTje experiment for level 1 categories

The fine-tuned BERTje experiment scored the highest for all metrics and all layers in the taxonomy with an MCC score of 95.40% for the lowest level in the taxonomy. The variation in performances could be due to several reasons. One advantage of BERTje over the other pre-trained models is that it breaks up words into several chunks if words are not recognized. The term 'beerlager' for example occurred in 41 product descriptions. The tokenizer function

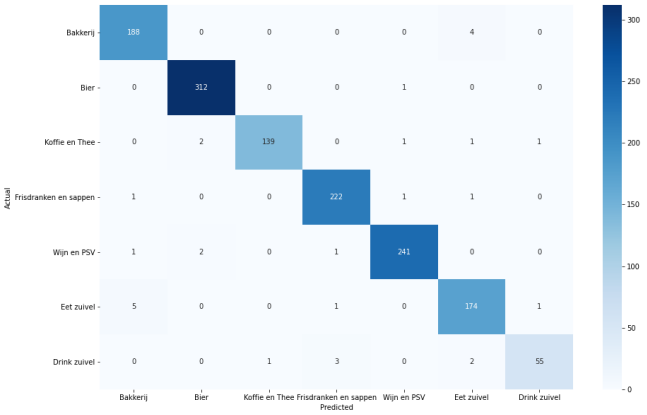


Figure 7: Confusion matrix of the BERTje experiment for level 2 categories

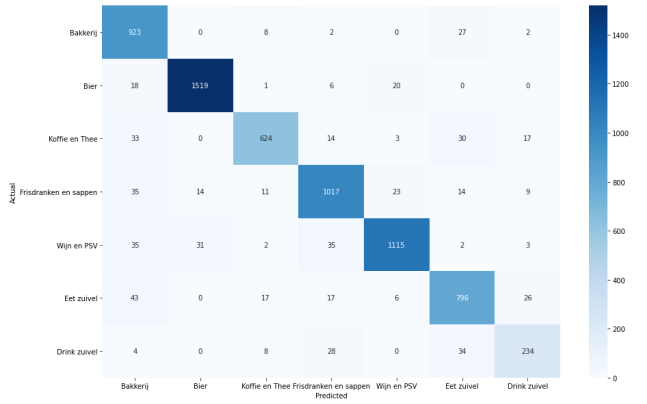


Figure 8: Confusion matrix of the TF-IDF experiment for level 2 categories

of BERTje was able to split the word into two recognizable words ('beer' and 'lager'). Another reason could be the different classifiers used: the BertForSequenceClassification and XGBoost classifier.

False predictions can be made due to ambiguity. The ground truth labels are created and assigned to the products by employees causing the involvement of subjectivity. Products within different categories may be close to each other in terms of similarity causing ambiguity. Therefore, the incorrect predictions may provide new perspectives on the categories assigned to particular products. An example of such a new perspective is given below. A product is assigned to the 'custard' category. However, words such as "dessert" and "raspberry" indicate that the product may also fit into the 'other eat dairy' category (which consists mainly of dessert products).

5.3 Limitations

The study has a number of limitations. One limitation is that the scope of products was kept limited due to the quality of the labeled dataset. This led to the exclusion of certain products of Albert Heijn's assortment in the study. The limited scope resulted in less training and evaluation data being available than expected. Stratified KFold was used to make the best use of the available data to contribute to the validity of the study. In terms of scalability, BERTje has proven to be valuable when predicting 29 categories in the third taxonomy level. However, increasing the complexity of the taxonomy by adding a fourth layer or multiple paths to one product, will require extra pre-processing and training.

Another limitation is the use of the XGBoost classifier as a baseline method. Neural networks are perhaps more suitable for unstructured textual data. However, the XGBoost classifier proved to be a solid baseline model for accurately predicting the product categories.

In addition, this research focused on the categories per level and ignored their hierarchical structure. The fine-tuned BERTje model is not able to classify by node in the product taxonomy. This was not investigated in depth because the focus of this study was the role of word embeddings in a product classification task.

The supplier descriptions are raw and thus have not been checked for quality and correctness. Some errors made by fine-tuned BERTje were caused by the use of misleading words in these descriptions (table 6). In the interest of a clear explanation, the descriptions have been translated into English.

Description	Actual	Predicted
beer san miguel e**special** 33 cl bottle	Lager	Specialty beer
wht white wine four cousins skinny red	Red wine	White wine

Table 6: Examples of misleading descriptions

In terms of generalisability, the findings of this study should be interpreted carefully. Some degree of subjectivity is inherent in the labeled products in the dataset. Products were labeled by data specialists from Albert Heijn based on their own perceptions and experiences with the products. Using a dataset labeled by other individuals (e.g. data from a different retailer) may not lead to the same results.

6 CONCLUSION

With the increasing popularity of e-commerce, retailers are being forced more and more to optimize and automate their data processes. This includes automating the classification of products into the right categories. Studies have shown that BERT can achieve promising results. However, to the best of our knowledge, no studies have shown the capability of BERT trained on the Dutch language to classify products based on their description.

This research aims to answer the question if BERTje could improve the classification of products when compared to the other Dutch pre-trained models Word2vec and FastText. The results show that for the product classification task BERTje outperformed other Dutch pre-trained models. The fine-tuned BERTje model obtained the best scores for all levels in the taxonomy. The model scored an MCC score of 98.03% for the first level, 98.16% for the second level and 95.40% for the third level. The results of our model showed that the product taxonomy consists of ambiguous categories, wherein products can and should belong to multiple categories. This highlights the importance of a taxonomy where products are multi-labelled.

We also concluded that simple word embedding such as TF-IDF techniques should not be underestimated in a classification task and are still effective depending on the data. Pre-trained word embedding models like fastText and Word2vec, could perform better when the corpus has a wider variety of words.

6.1 Future work

Future research could focus on a expanded product taxonomy. This research focused on a particular scope within the product assortment of Albert Heijn due to the quality of the labelled dataset. Products that fell outside the scope would later require labeling and thorough checking by professionals. This research has shown that the BERTje model promising results for this dataset. Therefore, the same approach can be taken in the future on a dataset with more categories and more product taxonomy levels.

More categories implies more types of products. When the data set expands adding more labeled products it also means that the corpus contains more words. Instead of using pre-trained models, fastText, Word2vec or GloVe embedding models could be trained on the available corpus. An advantage of training the embedding is that the models recognize domain specific words which can lead to an improvement in the classification products.

This research focused on a taxonomy where products can only belong to one category at each level in the taxonomy (single labelled). Future research could evaluate the performance of BERTje when products are multi labelled. This would be interesting since the BERTje model will be more complex because it has to classify per level in the taxonomy. Multiple paths in the taxonomy may lead to the same product. Irrelevant paths should therefore be excluded for particular products.

REFERENCES

[1] [n.d.]. *GS1 identification keys | GS1.* https://www.gs1.org/standards/id-keys

[2] Barbara Baarsma and Jesse Groenewegen. 2021. COVID-19 and the Demand for Online Grocery Shopping: Empirical Evidence from the Netherlands. *De Economist* 169, 4 (2021), 407–421.

[3] Ye Bi, Shuo Wang, and Zhongrui Fan. 2020. A Multimodal Late Fusion Model for E-Commerce Product Classification. *ArXiv* abs/2008.06179 (2020).

[4] Federico Bianchi, Bingqing Yu, and Jacopo Tagliabue. 2021. BERT Goes Shopping: Comparing Distributional Models for Product Representations. *ArXiv* abs/2012.09807 (2021).

[5] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2016. Enriching Word Vectors with Subword Information. *arXiv preprint arXiv:1607.04606* (2016).

[6] Vishwa Chandra, Prabh Gill, Kumar Venkataraman, Janice Yoshimura, and Varun Mathur. 2022. The next horizon for grocery e-commerce: Beyond the pandemic bump. https://www.mckinsey.com/industries/retail/our-insights/the-next-horizon-for-grocery-ecommerce-beyond-the-pandemic-bump

[7] Hongshen Chen, Jiashu Zhao, and Dawei Yin. 2019. Fine-Grained Product Categorization in E-commerce. *Proceedings of the 28th ACM International Conference on Information and Knowledge Management* (2019).

[8] L. Chen, Houwei Chou, Yandi Xia, and Hirokazu Miyake. 2021. Multimodal Item Categorization Fully Based on Transformer. In *ECNLP*.

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

[10] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *ArXiv* abs/1810.04805 (2019).

[11] Murhaf Fares, Andrey Kutuzov, Stephan Oepen, and Erik Velldal. 2017. Word vectors, reuse, and replicability: Towards a community repository of large-text resources. In *NODALIDA*.

[12] Ginny Fisher, Joe Brandenburg, Kees Jacobs, Norman Rosenberg, Srikant Kanthadai, Sunitha Ray, and Vaibhav Kumar. 2017. The devil is in the product data details - capgemini.com. https://www.capgemini.com/wp-content/uploads/2017/07/the_devil_is_in_the_product_data_details_.pdf

[13] Michał Graczyk and Kevin Kinningham. 2015. Automatic Product Categorization for Anonymous Marketplaces.

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

[15] Yonatan Hadar and Erez Shmueli. 2021. Categorizing Items with Short and Noisy Descriptions using Ensembled Transferred Embeddings. *ArXiv* abs/2110.11431 (2021).

[16] Hadi Jahanshahi, Ozan Ozyegen, Mucahit Cevik, Beste Bulut, Deniz Yiğit, Fahrettin Firat Gonen, and Ayse Basar. 2021. Text Classification for Predicting Multi-level Product Categories. *ArXiv* abs/2109.01084 (2021).

[17] Kamran Kowsari, Donald E. Brown, Mojtaba Heidarysafa, K. Meimandi, Matthew S. Gerber, and Laura E. Barnes. 2017. HDLTex: Hierarchical Deep Learning for Text Classification. *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)* (2017), 364–371.

[18] Kamran Kowsari, K. Meimandi, Mojtaba Heidarysafa, Sanjana Mendu, Laura E. Barnes, and Donald E. Brown. 2019. Text Classification Algorithms: A Survey. *Inf.* 10 (2019), 150.

[19] Abhinandan Krishnan and Abilash Amarthaluri. 2019. Large Scale Product Categorization using Structured and Unstructured Attributes. *ArXiv* abs/1903.04254 (2019).

[20] Maggie Yundi Li, Stanley Kok, and Liling Tan. 2018. Don't Classify, Translate: Multi-Level E-Commerce Product Categorization Via Machine Translation. *arXiv preprint arXiv:1812.05774* (2018).

[21] Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. 2013. Efficient Estimation of Word Representations in Vector Space. In *ICLR*.

[22] Ozan Ozyegen, Hadi Jahanshahi, Mucahit Cevik, Beste Bulut, Deniz Yiğit, Fahrettin Firat Gonen, and Ayse Basar. 2022. Classifying multi-level product categories using dynamic masking and transformer models. *Journal of Data, Information and Management* 4 (2022), 71 – 85.

[23] Sejoon Park, Chul-Ung Kang, and Yungcheol Byun. 2021. Extreme Gradient Boosting for Recommendation System by Transforming Product Classification into Regression Based on Multi-Dimensional Word2Vec. *Symmetry* 13 (2021), 758.

[24] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. GloVe: Global Vectors for Word Representation. In *EMNLP*.

[25] Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. In *NAACL*.

[26] Rukhma Qasim, Waqas Haider Bangyal, Mohammed A. Alqarni, and Abdulwahab Ali Almazroi. 2022. A Fine-Tuned BERT-Based Transfer Learning Approach for Text Classification. *Journal of Healthcare Engineering* 2022 (2022).

[27] Virginia Simmons, Julia Spielvogel, Björn Timelin, and Madeleine Tjon Pian Gi. 2022. The next S-curve of growth: Online grocery to 2030. https://www.mckinsey.com/industries/retail/our-insights/the-next-s-curve-of-growth-online-grocery-to-2030

[28] Koustuv Sinha, Yue Dong, Jackie Chi Kit Cheung, and Derek Ruths. 2018. A Hierarchical Neural Attention-based Text Classifier. In *EMNLP*.

[29] Chi Sun, Xipeng Qiu, Yige Xu, and Xuanjing Huang. 2019. How to Fine-Tune BERT for Text Classification?. In *CCL*.

[30] Liling Tan, Maggie Yundi Li, and Stanley Kok. 2020. E-Commerce Product Categorization via Machine Translation. *ACM Transactions on Management Information Systems (TMIS)* 11 (2020), 1 – 14.

[31] Parth Vora, Mansi Khara, and Kavita Kelkar. 2017. Classification of Tweets based on Emotions using Word Embedding and Random Forest Classifiers. *International Journal of Computer Applications* 178 (2017), 1–7.

[32] Pasawee Wirojwatanakul and Artit Wangperawong. 2019. Multi-Label Product Categorization Using Multi-Modal Fusion Models. *ArXiv* abs/1907.00420 (2019).

[33] Yonghui Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason R. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation. *ArXiv* abs/1609.08144 (2016).

[34] Li Yang, E. Shijia, Shiyao Xu, and Yang Xiang. 2020. Bert with Dynamic Masked Softmax and Pseudo Labeling for Hierarchical Product Classification. In *MWPD@ISWC*.

[35] Lina Yang, Ying Yang, Huanhuan Yu, and Guichun Zhu. 2019. Anonymous market product classification based on deep learning. In *AIIPCC '19*.

[36] Hsiang-Fu Yu, Chia-Hua Ho, Prakash Arunachalam, Manas Somaiya, and Chih-Jen Lin. 2012. Product Title Classification versus Text Classification.

[37] Hamada M. Zahera and Mohamed Ahmed Sherif. 2020. ProBERT: Product Data Classification with Fine-tuning BERT Model. In *MWPD@ISWC*.

A CATEGORY TRANSLATIONS

Level 1		Level 2		Level 3	
Dutch	*English*	*Dutch*	*English*	*Dutch*	*English*
Markthal & Bakkerij	Marketplace & Bakery	Bakkerij	Bakery	Grootbrood	Largebread
				Stokbrood	Baguette
				Zoet en gevuld brood	Sweet and stuffed bread
				Klein brood	Smallbread
				Gebak	Pastry
				Koek	Cookies
Dranken	Beverages	Bier	Beer	Bier pils	Pilsner
				Speciaal bier	Special beer
		Wijn & PSV	Wine & PSV	Witte wijn	White wine
				Rode wijn	Red wine
				Rose	Rose
				Apperitieven & mixdranken	Aperitifs & mixed drinks
		Frisdranken & sappen	Sodas & juices	Water	Water
				Fruit en groenten sappen	Fruit and vegetable juices
				Siropen	Syrups
				Frisdranken	Sodas
		Koffie & thee	Coffee & tea	Koffie	Coffee
				Thee	Tea
				Koffie & thee benodigdheden	Coffee and tea supplies
Zuivel	Dairy	Eetzuivel	Eatdairy	Kookzuivel	Cooking dairy
				Vla	Custard
				Yoghurt	Yoghurt
				Kwark	Quark
				Kaas	Cheese
				Boter & margarine	Butter & margarine
		Drinkzuivel	Drinking dairy	Drinkyoghurt	Yoghurt drinks
				Melk	Milk
				Ijskoffie	Iced coffee

Table 7: Taxonomy structure with English translations

B CATEGORY DISTRIBUTION

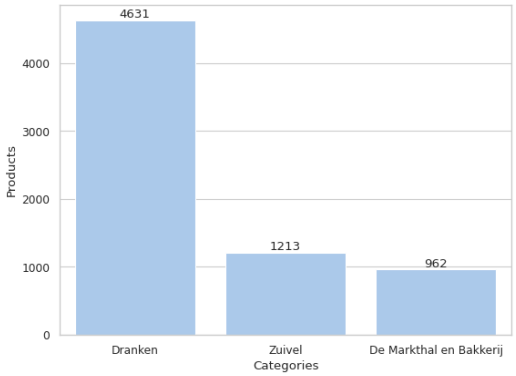


Figure 9: Distribution of categories of level 1

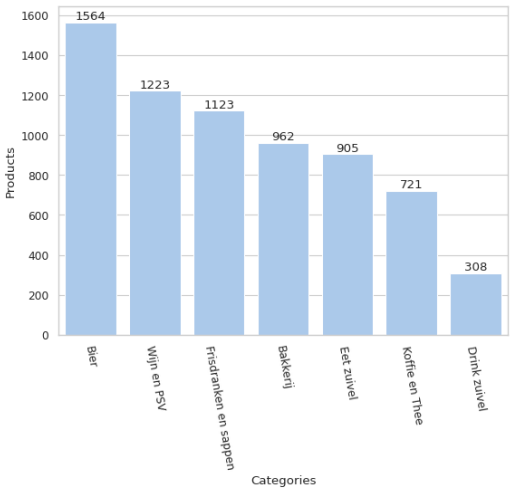


Figure 10: Distribution of categories of level 2

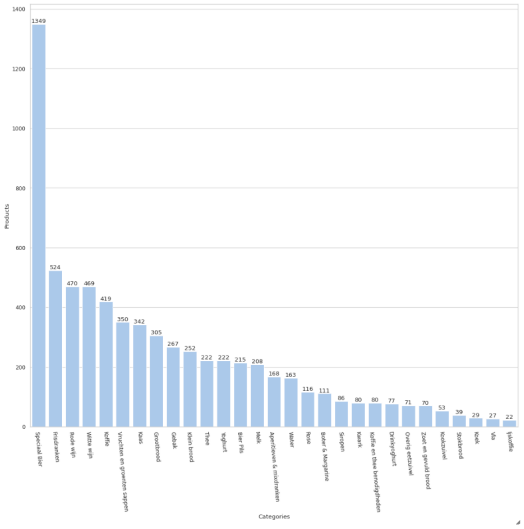


Figure 11: Distribution of categories of level 3

C WORDCLOUDS FOR HIGHEST TAXONOMY LEVEL

Figure 12: Wordcloud of bakery category

Figure 13: Wordcloud of beverages category

Figure 14: Wordcloud of dairy category

D VISUALIZATIONS OF PRODUCT REPRESENTATIONS

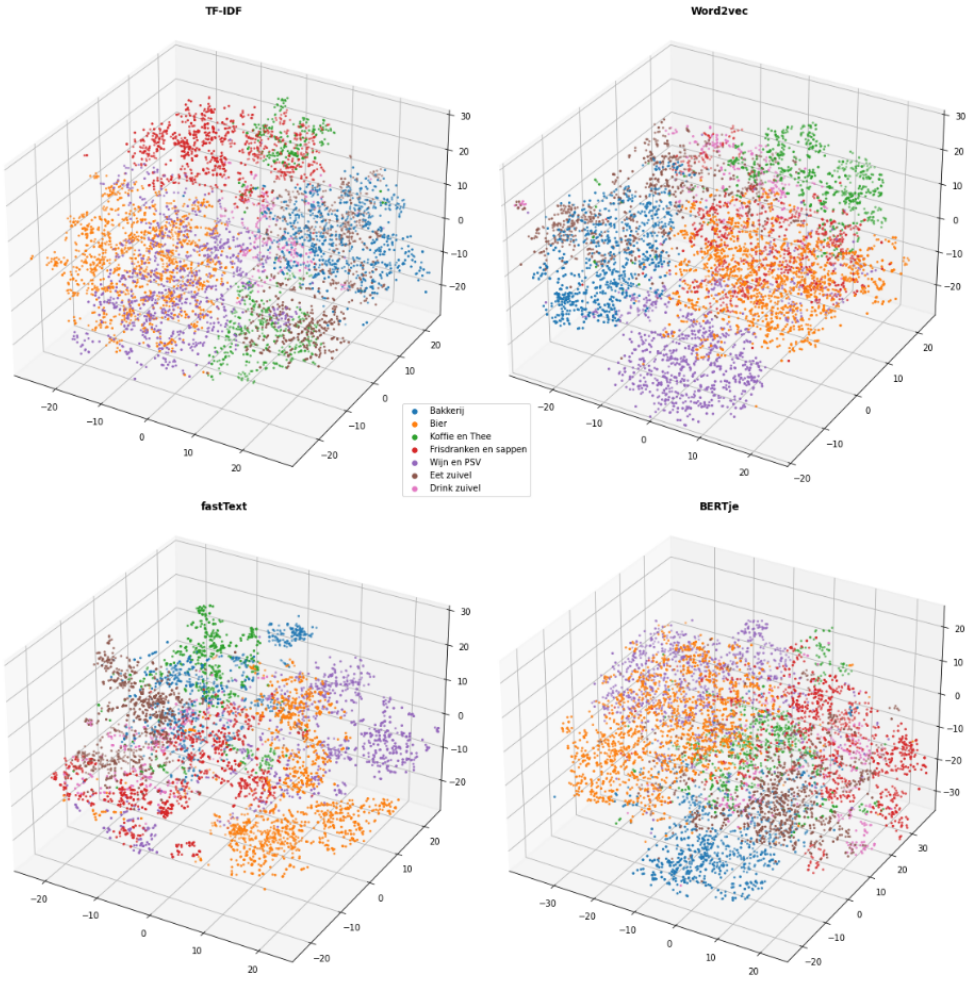


Figure 15: Product embeddings with level 2 categories as labels

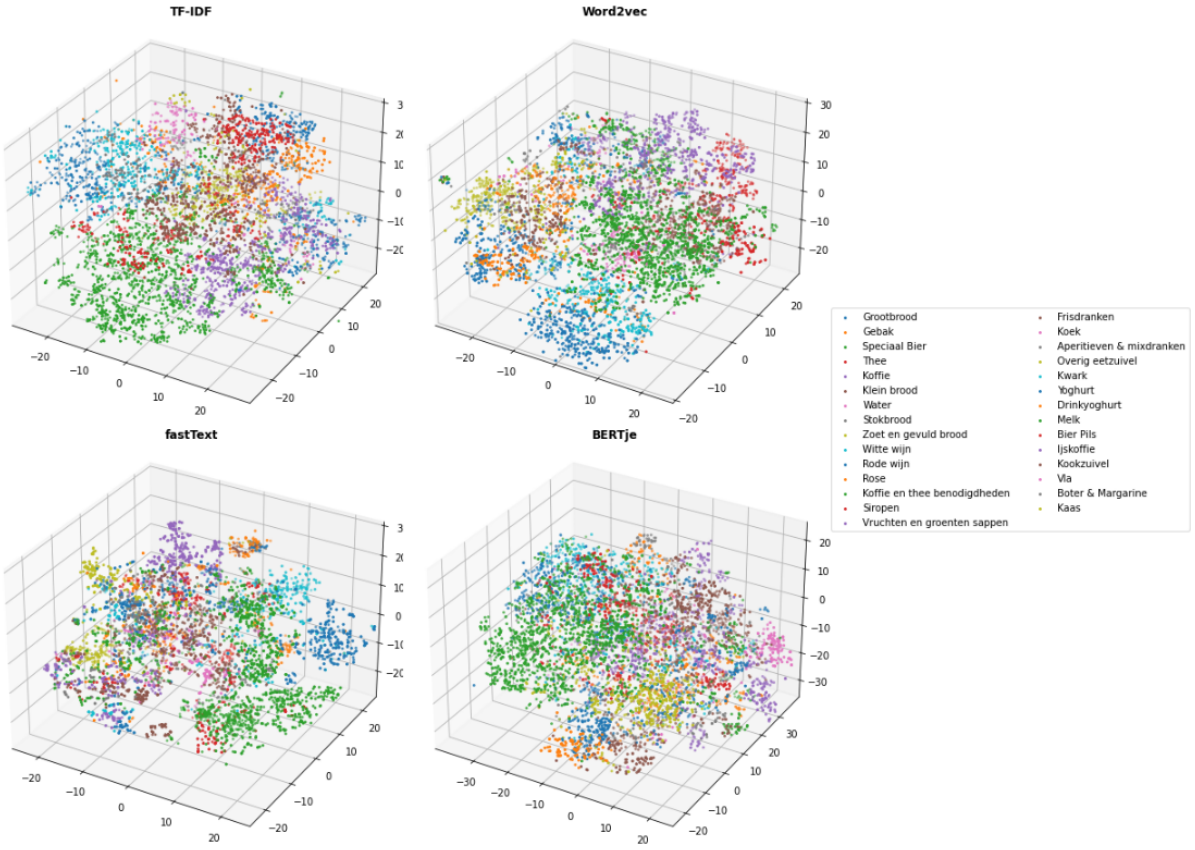


Figure 16: Product embeddings with level 2 categories as labels

E TRAIN AND VALIDATION LOSS

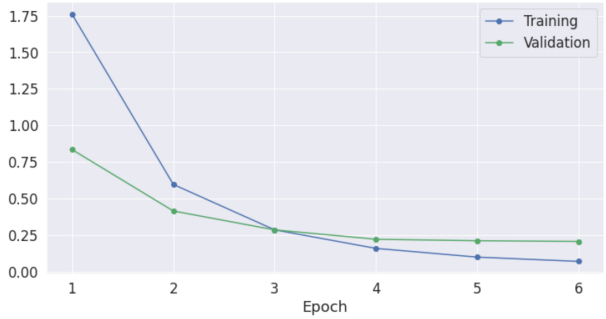


Figure 17: Train en validation loss per epoch for BERTje

F CONFUSION MATRICES

F.1 TF-IDF experiment

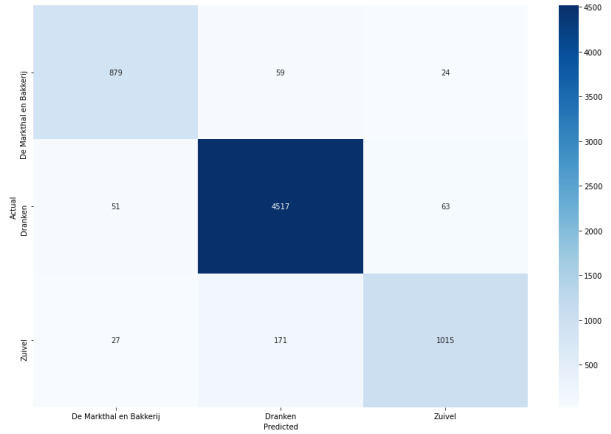


Figure 18: Confusion matrix of the TF-IDF experiment for level 1

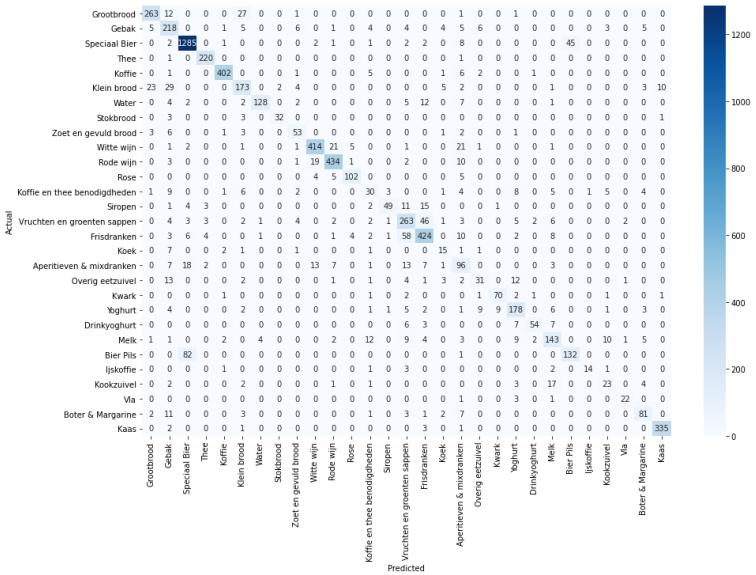


Figure 19: Confusion matrix of the TF-IDF experiment for level 3

F.2 Word2vec experiment

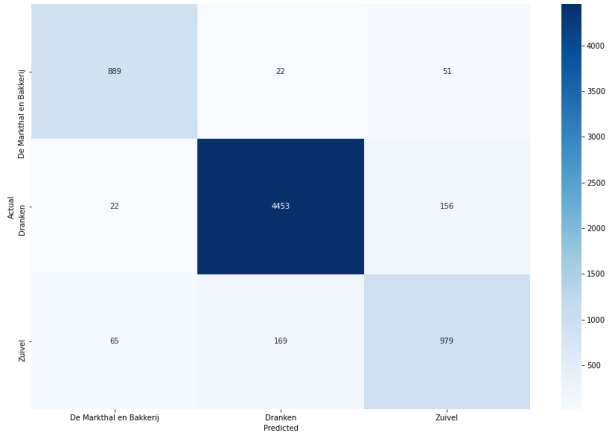


Figure 20: Confusion matrix of the Word2vec experiment for level 1

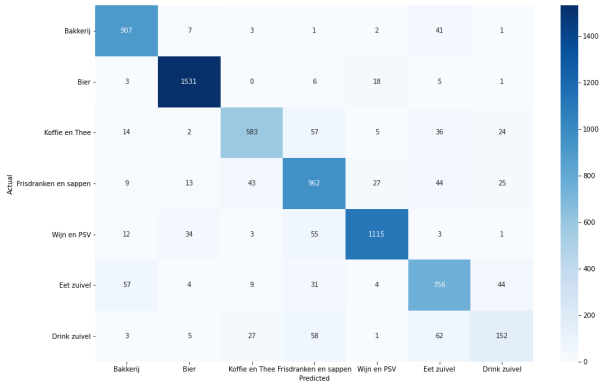


Figure 21: Confusion matrix of the Word2vec experiment for level 2

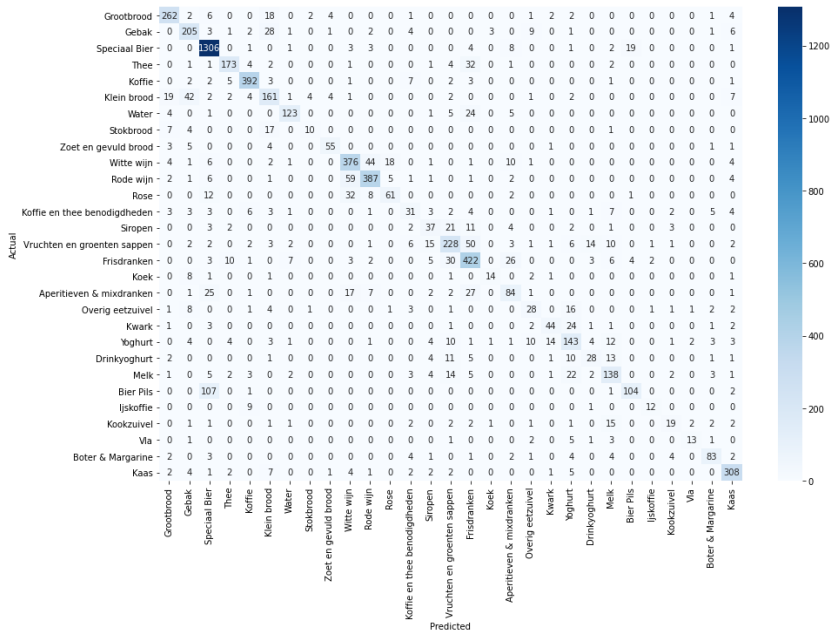


Figure 22: Confusion matrix of the Word2vec for level 3

F.3 fastText experiment

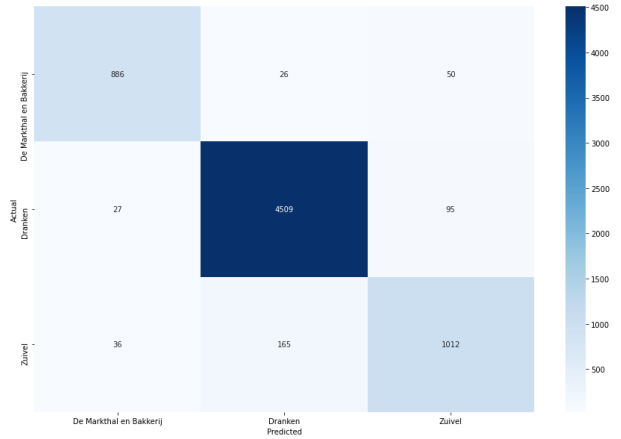


Figure 23: Confusion matrix of the fastText experiment for level 1

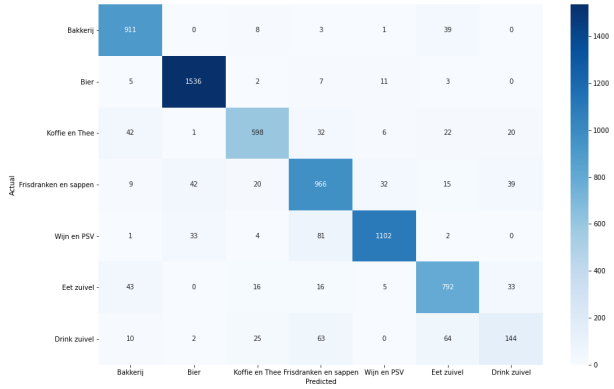


Figure 24: Confusion matrix of the fastText experiment for level 2

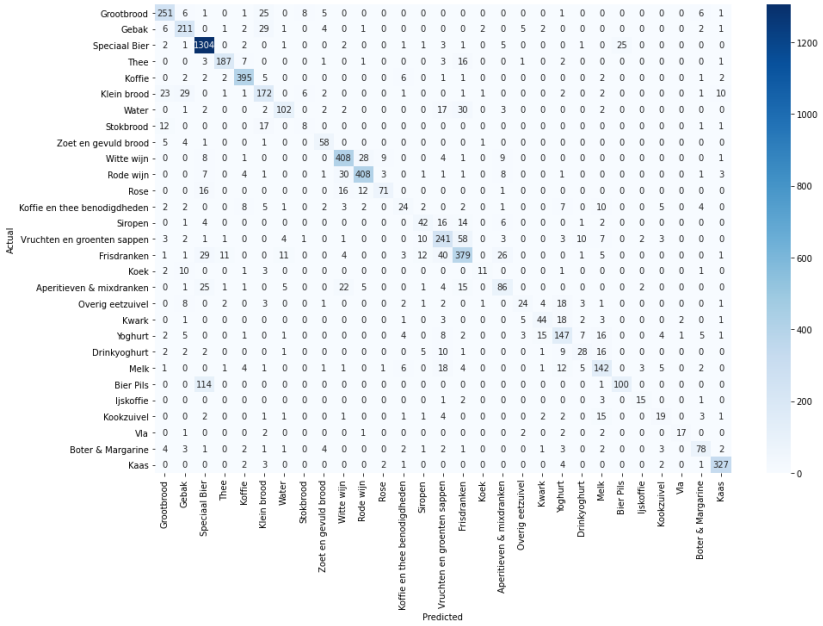


Figure 25: Confusion matrix of the fastText for level 3

F.4 BERTje experiment

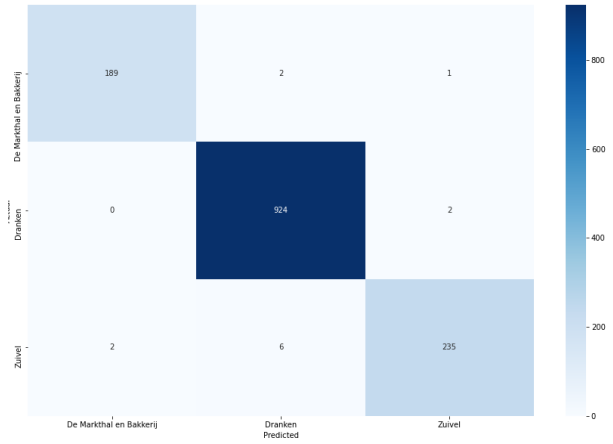


Figure 26: Confusion matrix of the BERTje experiment for level 1

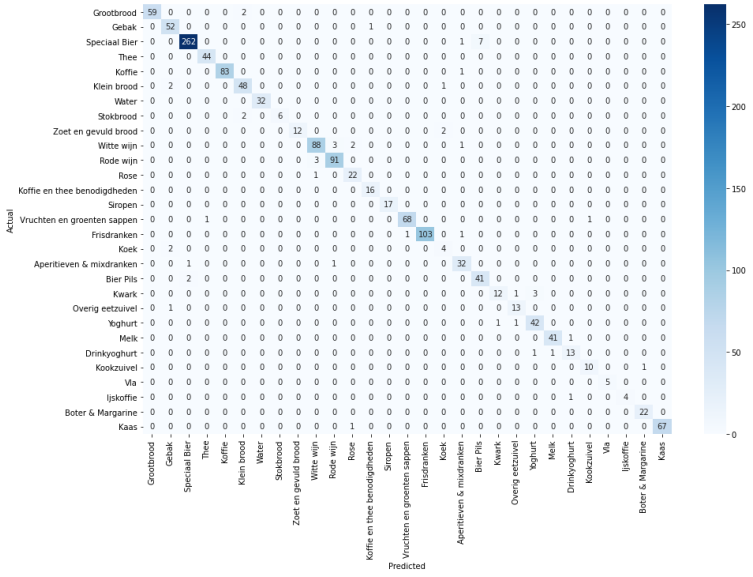


Figure 27: Confusion matrix of the BERTje for level 3