A Deep Learning Approach to Interest Analysis

Thomas van der Meer

A thesis submitted for the degree of
Master of Business Informatics

Department of Information and Computing Sciences
Utrecht University
The Netherlands

13th of July 2020

Abstract

The analysis of interests from young adolescents in the form of short, colloquial Dutch text is a challenging task for pre-trained neural networks. By quantitative and qualitative tests, four pre-trained language models on the Dutch language are compared and contrasted. Three more language model fine-tuned models are added to test transfer learning capabilities for the qualitative tests. By training a classifier on a named entity recognition- and sentiment analysis task, the models are quantitatively compared. For the qualitative comparison, The outputs from the embedding layer are used to gain insight in relation classification and clustering. A test for ranking interest pair similarities has been developed in order to investigate the semantical understanding of the Dutch language in the models. Furthermore, the clustering capabilities of related interests are examined. Finally, given relation structures in sports, instruments and school courses are brought to a test. BERTje outperforms the other models in the quantitative tasks. However, BERTje performs the worst on the triplets ranking test. RobBERT fine-tuned and FastText show the best results on the triplets analysis. All models lack to show semantical understanding in the clustering analysis. FastText shows the most semantical understanding in the relation structures, though still relatively poor. The outputs from the embedding layer shows that the models do not have a semantical understanding of the Dutch language but fall back on morphological structures. Therefore, these techniques are not ready to be used for interest analysis. Creating a downstream task, data enrichment and knowledge infusion are candidates for improvements on interest analysis.

Dedication

To my mum and dad, who throughout my life always have been an inspiration and example. They, together with my sister, created a family to always love to come back to and spent precious time with.

To Laura, the one that day in and day out stood by me. Together has celebrated my victories and endured my setbacks.

Acknowledgements

I would like to thank both my supervisors, Marco Spruit and Sanne Akkerman, for their guidance through these invigorating months of working on my thesis. You both have helped me tremendously by always thinking critically and put immeasurable amounts of energy in the theoretical part of this work. Furthermore, I would like to thank Jael Draijer and Joris Beek for their help with the analysis of the qualitative reviews, being there as sounding board and advisors, and most importantly, the comradery at the office and from home. Lastly, Thank you Pablo Mosteiro for your insights in NLP techniques and your technical perspective on all this.

Contents

List of Figures

List of Tables

list of acronyms and abbreviations

AR	autoregressive
BERT	Bidirectional Encoder Representations from Transformers
biLM	bidirectional language model
CBOW	continuous bag-of-words
CoLA	Corpus of Linguistic Acceptability
DBRD	Dutch Book Review Database
ESM	experience sampling method
GPT	Generative Pre-Training
LM	language modelling
LSA	latent semantic analysis
LSTM	long short term memory
MCC	Matthews correlation coefficient
MLM	masked language modelling
NER	named entity recognition
NLP	natural language processing
NLU	natural language understanding
OOV	out-of-vocabulary
PCA	principal component analysis
PRAW	Python Reddit API Wrapper
RNN	recurrent neural network
SOTA	state of the art
t-SNE	t- Distributed Stochastic Neighbor Embedding
UMAP	Uniform Manifold Approximation and Projection
110kDBRD	110k Dutch Book Review Dataset

Chapter 1

Introduction

The individual interests of people are unique and develop from situations throughout life (Akkerman & Bakker, 2019). In order to understand the interest development, daily activities of people are tracked to gain insight. The currently running research project titled “Lost in Transition: Multiple Interests in Contexts of Education, Leisure and Work” gathers this data with the goal to find out how different interests relate to each other and how these interests develop over time. This is done through an experience sampling method (ESM) data collection process, where events throughout the day are recorded by the user themselves. The nature of the data are short texts, written in colloquial Dutch, containing possible slang, misspellings and other contaminations. These properties provide an extra challenge. The task at hand is to analyse the data effectively through automated methods in order to ultimately map interest development of people over time.

The research landscape of natural language processing (NLP) has radically changed over the last decade. The ability to train word representation models with vast amounts of natural language (Mikolov, Sutskever, et al., 2013), has been a catalyst to a wide range of new techniques. Using unsupervised training of a neural network on large datasets, a sense of syntactical, semantical and contextual awareness can be found in the word representations (M. Peters et al., 2018). The downside to enormous models is the vast amount of time and resources needed to create a state of the art (SOTA) performing model. By fine-tuning, the so called ‘pre-trained models’ are used as starting point by providing the word representations and hereafter the word representations are used to perform different tasks, such as named entity recognition (NER), sentiment analysis and many more (M. E. Peters et al., 2017). These different tasks are called downstream tasks and translate a certain analytical goal to a process that an NLP model can perform. Not only classifier fine-tuning can be done, also language model fine-tuning (Howard & Ruder, 2018) is important to capture the idiosyncrasies of the target corpus. Adapting a language model to a certain target domain by feeding domain specific texts, is called transfer-learning (Howard & Ruder, 2018).

Pre-trained models have the ability to be fit to your own dataset and downstream task, making it possible to use the captured word information from SOTA models and fit this to your own dataset, task and language (Howard & Ruder, 2018)(M. Peters et al., 2018). The range of pre-trained models is vast, varying in input from character- (Akbik et al., 2018), to word- (Bojanowski et al., 2017) and sentence level (Devlin et al., 2018; Liu et al., 2019).

The models are pre-trained on enormous datasets of text, however most of the time only in English. There are multilingual models (Devlin et al., 2018) available and native Dutch pre-trained models (de Vries et al., 2019; Delobelle et al., 2020). Unfortunately, not trained on as much data and graphical computer power as their English counterparts. A qualitative comparison of Dutch models to find out the current SOTA will be helpful to determine the feasibility of text analysis in Dutch.

This thesis contributes to the applicability of NLP methods in a real-world case. Furthermore, the used techniques are evaluated and compared to each other and try to uncover the constraints of using NLP in this domain and language. Lastly, a tool is developed to give researchers the opportunity to use the models themselves and the ability to create analyses on gathered data.

1.1 Research approach

1.1.1 Research goal

The research aim for this thesis is to gain insight in how to apply NLP and machine learning techniques in order to deal with large amounts of moment-to-moment interest experiences of adolescents over time. In greater detail, the goal is to establish an analysis and understanding of the current NLP- and machine learning techniques applicable for this problem.

1.1.2 Research questions

Based on the research goal stated above (1.1.1) the following research question is asked: What NLP techniques can be applied and perform well in order to analyse Dutch interest data captured over time while accounting for colloquial language used by adolescents? This question leads to the following sub-questions, listed below.

- RQ1: How do different (pre-trained) NLP models relate to each other in terms of performance.
- RQ2: How to account in the pipeline for the use of the Dutch language in modelling personal interests?
- RQ3: What ways can the models be visualized and how interpret the results?

The goal of RQ1 is finding out in what ways and how well the different word representations can capture the semantics of the data in the dataset. This will be done through using the different models and create (contextualized) word and sentence representations, thereafter the representations will be reviewed and scored through a quantitative method and a qualitative review. To nuance the performance in the research question; Performance is measured through metrics on the quantititave methods and through qualitative analysis in the qualitative method. The focus in the qualitative method is the interest data provided by the research group and therefore the models are evaluated in how they grasp Dutch interests.

In relation to RQ2, To address the more practical side of creating NLP models, a rigour set of tools is needed to create a pipeline especially for analysing Dutch interest data. For example, there are models available trained on Dutch corpora (de Vries et al., 2019; Delobelle et al., 2020).

Finally, RQ3 is a more practical question of this thesis that is designed to create a tool to evaluate and review the representations and the interest development over time.

1.1.3 Research method

For this thesis, the research will be conducted through the lens of the engineering cycle (Wieringa, 2014). The engineering cycle depicts a rigid research method based on design science, aimed at the field of information and software engineering. By following the engineering cycle, 4 different phases will be touched on, namely Problem investigation, Treatment design, Treatment validation and Treatment implementation. Figure 1.1 displays the engineering cycle. Key is that the object of study is the artefact in context, in this case comparing the models based on the performance on interest data (Wieringa, 2014). This case-based research will give new insights from the observed behaviour of the artefact in context and hopefully be generalized, or at least provide a guide on how NLP can be researched and applied in a real-world context.

For this study, the Problem investigation, Treatment design and Treatment validation stages will be filled out, in order to get a structured overview of the research. The experiment design is elaborated on including the stages and the research questions it addresses.

Problem investigation

As mentioned before, but to state nevertheless, the problem for this research entails around the notion of digital colloquial Dutch that needs to be analysed. The stakeholders are the research group that focusses on the research "Lost in Transition: Multiple Interests in Contexts of Education, Leisure and Work". The stakeholder's goals are aligned in doing research into text analysis using NLP techniques. The question is if the models have the ability to relate interest data in a meaningful way and explore the ability of the models to uncover relations the researchers cannot find.

To delve deeper into the matter, a literature review is conducted to find out the

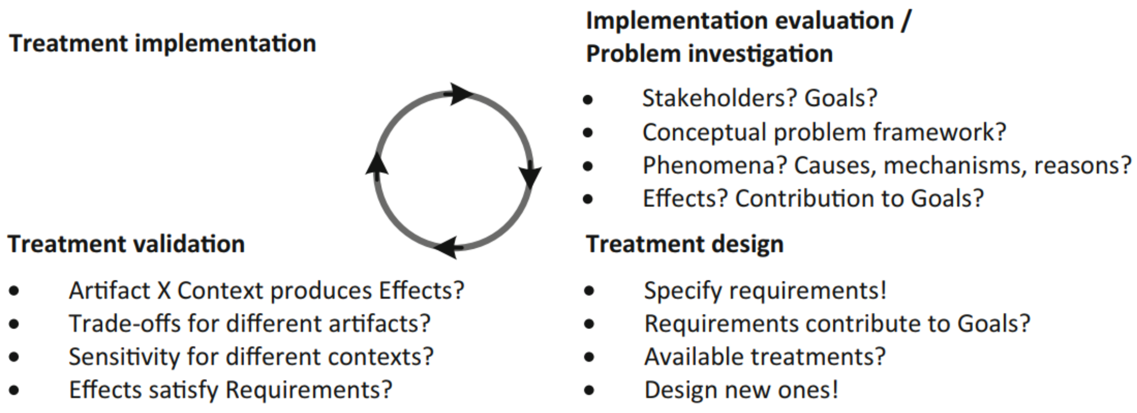

Treatment implementation
Implementation evaluation /
Problem investigation
• Stakeholders? Goals?
• Conceptual problem framework?
• Phenomena? Causes, mechanisms, reasons?
• Effects? Contribution to Goals?
Treatment validation
• Artifact X Context produces Effects?
• Trade-offs for different artifacts?
• Sensitivity for different contexts?
• Effects satisfy Requirements?
Treatment design
• Specify requirements!
• Requirements contribute to Goals?
• Available treatments?
• Design new ones!

Figure 1.1: Engineering cycle (Wieringa, 2014)

current state of NLP research, the common ground and the new findings of the last decade. The literature review can be found in chapter 2.

Treatment design

The treatment design is a step of working together with the ERC researchers in order to translate the requirements and results from the problem investigation into a treatment that captures the essence of the research questions. Treatment availability is there to a certain extent, but has to be brought together to make it whole. This means the availability of NLP models and packages to configure interactive tools.

Unfortunately, this is not an all-in-one product yet. The artefact is the NLP model differentiating in architecture, training and such, to find out which produces the best results for the quantitative and qualitative tests. The method of these tests can be found in chapter 3.

Treatment validation

To see if the treatment shows the desired effects, both qualitative and quantitative studies will be used. In more detail, the performance of the different models will be measured on a general performance benchmarking tests. In addition, the interest relation and categorisation will be scored on a qualitative basis. Lastly, the tool should perform well and be user-friendly to make sure that it will be used to do the analysis. The final artefact will be an interactive tool that helps to interact and analyse the data available. The results of the experiment will be reported in chapter 4.

1.2 Thesis outline

To reiterate, this thesis is ordered as follows. Chapter 2 will be a related works section on the inner workings of neural networks, the current state of neural networks in relation to NLP, evaluation of these methods and miscellaneous subjects that relate closely to the research and are important for further, deeper understanding of the methods in this thesis. The next chapter, chapter 3, will be an overview of the method for conducting the research where the treatment design is central. Chapter 4 will lay out the model analysis, describing the treatment validation process and subsequently the results. Finally, chapter 5 will be a combination of conclusions, discussion, limitations and further research.

Chapter 2

Related Works

The main goal of the literature review is to explore the different NLP models that are available and form the SOTA in NLP and natural language understanding (NLU). By chronologically walking through every model, the goal is to gain an understanding of the development of NLP, the different approaches there are to this multi-faceted problem and the sheer volume of work that is being put into this field to ultimately understand language as humans. To properly demonstrate the models in this literature review, there is knowledge needed about neural networks and the different ways a neural network can be altered to increase performance. Furthermore, the literature on transfer learning, ensemble methods and other subjects deemed important to better understand NLP models are introduced (2.1). Thereafter, the models are chronologically introduced, therefore mixing character-, word and sentence level models (2.2).

Lastly, section 2.4 will focus on pointing out the literature gap that exists and state the scientific contribution that is this thesis. By the broad beginning and the narrow ending of the literature review, the goal is that it states the collected knowledge of the field and subject, framing the thesis in the right perspective and finally the literature fundamentals that is needed to correctly conduct the research. The literature research protocol is provided in appendix A.

2.1 Language modelling

2.1.1 Definition of language modelling

The fundament of NLP is language modelling (LM). To define LM, "LM is a central task to NLP, where the goal is to learn a probability distribution over sequences of symbols pertaining to a language."(Jozefowicz et al., 2016). This has been done through different methods, such as a parametric model, count-based and since the current decade more through neural networks. To put into context, a five-gram (probability over five words) model from 1995 has been a strong baseline that has been competitive with neural network approaches (Jozefowicz et al., 2016). However, through new breakthroughs of machine learning, larger (annotated) datasets and more computing powers, model architectures are close or even outperforming human baselines on certain NLP-tasks (A. Wang, Pruksachatkun, et al., 2019).

2.1.2 Different kinds of neural networks

For the coming sections where the different architectures will be discussed, from Word2Vec 2.2.1 to T5 2.2.12, it is suitable to have a general understanding of neural networks, the fundamentals for these techniques and how they generally work. The exact implementation, activation functions, learning rates and other technical terms are not that important for now and will be discussed at implementation, but an intermediate understanding will most probably help in the coming sections.

2.1.3 Neural networks

A neural network is a mathematical function that maps a given input to a desired output. The simplest neural net is a 2-layer neural network where there is:

- An input layer (yellow), a hidden layer (blue) and an output layer (green).
- Between the layers, there are weights and biases that change the values of the last layer.

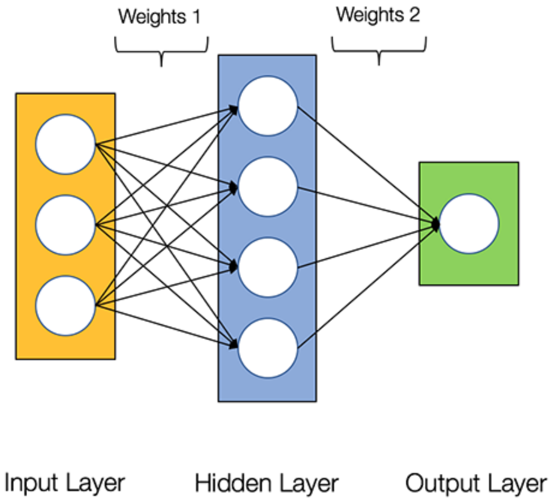


Figure 2.1: Neural network[1]

In order for an neural network to make predictions, the neural network has to be trained first. This is done with labelled data where the input goes through the neural network, calculating the predicted output, known as feedforward. Learning from the loss function, the difference between the actual and predicted output, is updating the weights and biases, known as backpropagation. By minimizing the loss function on a representative training dataset, a neural network can predict well. A neural network like this is schematically shown in figure 2.1.

2.1.4 Recurrent neural network

A recurrent neural network (RNN), as is used by Mikolov, Sutskever, et al. (2013) for Word2Vec (2.2.1), is a special type of neural network, where there is a link between the hidden layer to itself, therefore having an understanding what has been in the

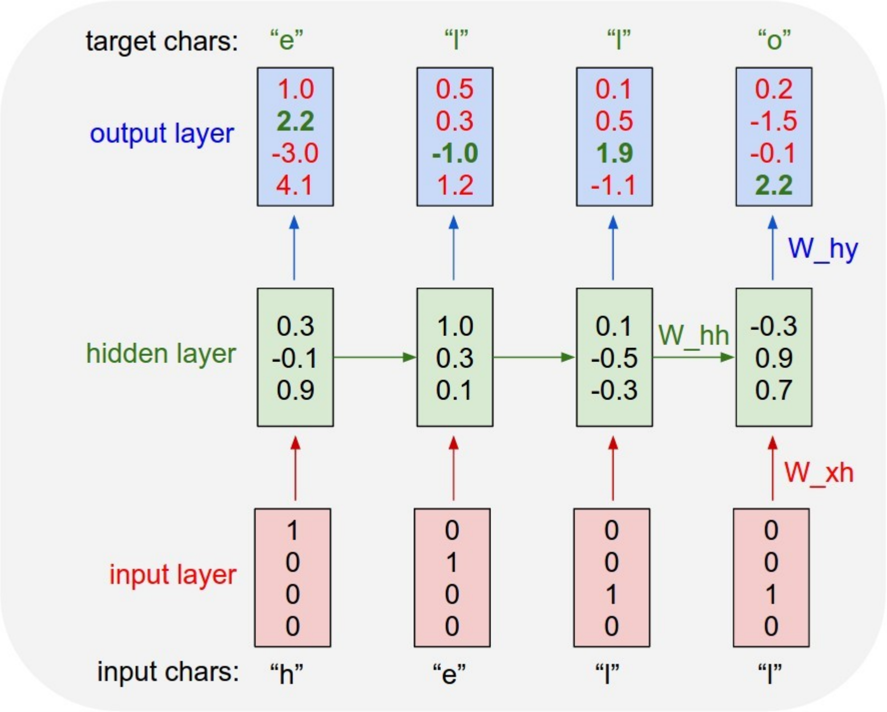


Figure 2.2: RNN neural network architecture[2]

input layer before. This kind of neural net was invented by (Rumelhart et al., 1986). For example, Figure 2.2 depicts a character-level model with the input characters (input chars) below, a vector representation in the input layer (red), a hidden layer (green) and an output layer (blue). As seen between the hidden units, there is a connection that shares information with the right unit, making the unit aware of the character before in the set.

To dive further into the example in figure 2.2, this RNN is a language model that tries to predict the next letter in the sentence. This example tries to spell “hello” based on the input chars “hell”. There are four possible letters, namely ‘h’, ‘e’, ‘l’ and ‘o’ and those are one-hot encoded into the input layers. The hidden layer is already trained and has the respective weights in the green boxes and outputs the score given in the blue box, the output layer, given the probability that it is likely an ‘e’ following the h. This is seen in the blue box on the left. The green number corresponds with the binary one-hot encoding. It gets interesting at the input of the second ‘l’, where the language model predicts not the same letter as outcome (‘l’) but the ‘o’, based on the RNN property of sharing the earlier inputs.

2.1.5 Long short term memory

Introduced by (Hochreiter & Schmidhuber, 1997), LSTMs are a solution to a shortcoming of RNNs. Long-term dependencies cannot be captured by RNNs, where long short term memory (LSTM)s are able to. While the idea is broadly similar, the LSTM network is designed to have a stable connection between all LSTM units, better at storing information from the other units and updating is only subtle. For example, (Howard & Ruder, 2018) used an LSTM for their, at that time, SOTA

[2] http://karpathy.github.io/2015/05/21/rnn-effectiveness/

language model. Note that this is only going from left to right and LSTMs are not bidirectional by definition.

2.1.6 Bidirectionality

Sharing information from earlier inputs has shown value in RNNs and LSTMs. However, what if you could learn from the later input? Bidirectionality was added to share information not only from left-to-right, meaning earlier input, but also from right-to-left (later input). Therefore, the neural network training can be done by using two neural nets that are in opposite direction and that 'combine' the two outcomes (M. Peters et al., 2018).

2.1.7 Transformer

The transformer is unlike LSTM, not a descendant of the RNN architecture, but works in a different kind of way. The transformer is the work of Vaswani et al. (2017). Below, in figure 2.3, there are two parts to a transformer, namely the encoder (denoted by the left Nx) and the decoder (the right Nx). The input is already different from the RNN and LSTM architecture, whereas the encoder is fed the entire sentence in contrary to only a word (or character as in Akbik et al. (2018)) in an RNN. The sentence is given an input embedding together with a positional encoding of each word. This is necessary in order to know where the words are in a sentence. The next step is the attention mechanism. Without going into much mathematical detail, the encoder creates key-value pairs that are remarkable to the sentence and gives those to the decoder. To simplify this, for every word in the sentence, a score is calculated that captures the importance of that word to the word in question and a matrix is created. The decoder has output embeddings that formulate what is needed from the input. For example, a LM task where the decoder has the previous word of the sentence as the output embedding and the next possible word as the input embedding. The decoder stack does the same trick for the output embedding, finding the remarkable part of the sentence until now, and passes this to the combining Multi-Head Attention block (With the arrows coming from the encoder). Here, the output with the remarkable parts of 'queries' for the possible input and the attention block looks for a key-value pair input embedding that would fit the task. This gives back some options and creates an output probability for the next word in the sentence. Transformers work well for NLP because unlike RNNs, transformers suffer less from great path-lengths on long range dependencies and can work in parallel (Vaswani et al., 2017).

2.2 Natural language processing state of the art

To elaborate on different techniques in order to understand written text, an exhaustive list of techniques will be discussed below. The focus will lie on the high-level functioning of the techniques, without diving too much into the math behind. The explanation will be detailed nevertheless. These explanations are mostly based on their own published articles and documentations.

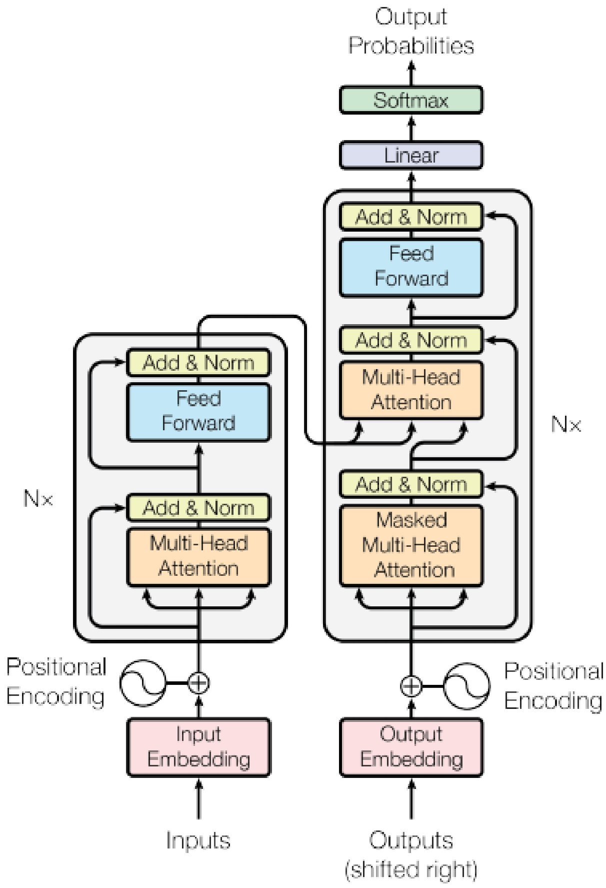


Figure 2.3: Transformer model architecture (Vaswani et al., 2017)

2.2.1 Word2vec

Word2vec, released in 2013, is a product out of the Google research lab where a breakthrough method was developed on creating a high-quality distributed vector representations in an efficient manner (Mikolov, Sutskever, et al., 2013). The idea is not new, whereas Collobert and Weston (2008) already proposed using a feed-forward neural network to learn word embeddings. For Mikolov, Sutskever, et al. (2013), the focus lies on the skip gram model, an architecture leveraging a neural network for learning word embeddings. In an earlier paper two new models have been introduced to learn word representations of large datasets, namely continuous bag-of-words (CBOW) and continuous skip-gram (Mikolov, Chen, et al., 2013). What makes this technique special, is the efficiency on learning word representations from large datasets and therefore an enabling technology for pre-trained word embeddings (Mikolov, Sutskever, et al., 2013).

Continuous Bag of Words

The CBOW architecture uses a feed-forward neural net (RNN) language model to essentially predict the target word based on the surrounding words. A projection layer averages the vectors of the surrounding words and uses this to predict the target word. Mikolov, Chen, et al. (2013) found that given the four words before and four words after the target word, the classifier would perform best in relation to the increasing computational complexity. Figure 2.4 (left) shows a diagram of how CBOW works.

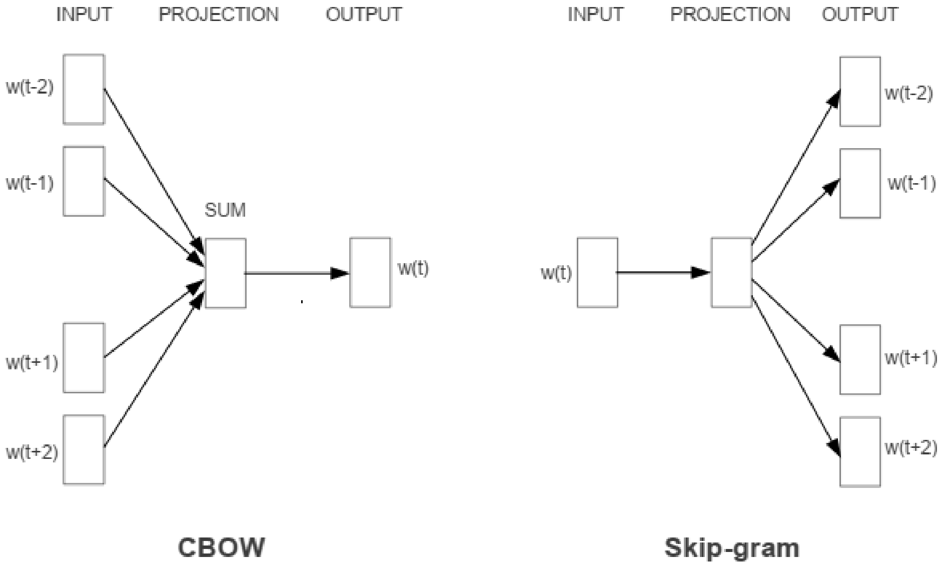


Figure 2.4: CBOW and Skip-gram architectures (Mikolov, Chen, et al., 2013)

Skip-gram

Skip-gram shares it similarity with the CBOW architecture mentioned above, but uses the current word in the sentence to predict the context. Every prediction is added to a log-linear classifier to predict words before and after the current word. Similar to CBOW, increasing the range of words that are predicted by the current word, improves the quality of the word vector but also increases the computational complexity. To account for the loss of relatedness of words that are further away in the context to the current word, the weight of surrounding words is higher than words that have a position further away in the context (Mikolov, Chen, et al., 2013). Figure 2.4 (right) shows the model architectures skip-gram. What Mikolov, Sutskever, et al. (2013) finally makes the wide adopted Word2Vec model are the following extensions;

- Subsampling of frequent words during training results in speedup and improves accuracy of less frequent words.
- - Additional use of phrase representations on top of word representations.

For training on large datasets, frequent words like “the” and “a” will very unlikely provide a meaningful relation with other words. To account for this, Mikolov, Sutskever, et al. (2013) use a simple subsampling formula (1 – the root of an arbitrary threshold divided by the frequency of the word) to discard the words whose have a frequency greater than an arbitrary amount, without losing the ranking of frequencies. This method has been found to accelerate the learning of the vectors and improve accuracy on rare words. For the optimization of word- to phrase representation, the decision fell on a simple data-driven approach. A large number of those phrases were identified and then used as individual tokens during training.

2.2.2 FastText

Bojanowski et al. (2017), working together with the first author of the Word2Vec model, pushed the performance of word embeddings with Fasttext, an improved architecture developed at Facebook. The initial release was in November 2015. When taking into account the Word2Vec technique, every word in the vocabulary is represented by a distinct vector.

However, many languages have many word forms like French verbs or Finnish nouns (Bojanowski et al., 2017). When these different word forms occur rarely in the training corpus, the model will not gain a thorough understanding of the language, and therefore make it difficult to learn good word representations.

Bojanowski et al. (2017) propose a solution by leveraging character information to possibly improve vector representation. This is done using character n-grams, and using these n-grams to represent words as the sum of the n-gram vectors. Note that this is an extension on the skip-gram model from Bojanowski et al. (2017).

For learning words, there are n-grams created from the words. An n-gram can be seen as an arbitrary part of a word. Bojanowski et al. (2017) use the example of the word “where”, where the word is represented by character n-grams of n = 3 including the full word. This results in the following representation of the word “where”: (wh, whe, her, ere, re) and (where). These are converted to vector representations and finally summed to create the final vector representation of the word “where”.

As expected, this extra step has influence on the performance of the model. In experimental setup, the FastText model performs approximately 1.5x slower than the skipgram baseline on English data. (105k words per second per thread vs 145k/words per second per thread) (Bojanowski et al., 2017).

Out of vocabulary words

What makes Fasttext even more interesting, is the ability to overcome the problem of out-of-vocabulary (OOV) words. The model is capable of creating word vectors for words that do not appear in the training set by using the n-grams vector representation that the word consists of and simply average those. Bojanowski et al. (2017) show using synonyms that a vector representation can be built on the n-grams of the OOV word that is roughly similar to the synonym. Bojanowski et al. (2017) use as examples microcircuit and chip, rarity and scarceness, where one word is in the training set and the other is not. This also indicates that prefixes and suffixes can be ignored for words that are not found in the vocabulary (Bojanowski et al., 2017).

2.2.3 GloVe

From the Stanford NLP group, mostly known for their widely used, integrated NLP toolkit Stanford CoreNLP, the GloVe model is introduced (Pennington et al., 2014). GLoVe differences from the other models by using both global and local statistics of the training corpus (Pennington et al., 2014). As described above, statistical methods over whole corpora have been standard for many years before Word2Vec. Methods like latent semantic analysis (LSA) and other matrix factorization methods create matrices of rows of words and columns of documents where words occur in context of another given word (Pennington et al., 2014). This is missing in the

Word2Vec model where only the surrounding words are used to predict the word in question (Pennington et al., 2014).

To explain the GloVe model, the researchers illustrate the relationships between words with the following example. Having the following words; ‘ice and steam’ with the following co-occurring words; ‘solid, gas, water and fashion.’ If we give a probability to the different words ‘ice’ will be related to ‘solid’ and ‘water’, minimal with ‘gas’ and not with ‘fashion’. For ‘steam’, this will be related to ‘gas’ and ‘water’, minimal with ‘solid’ and not with ‘fashion’. As we can see, both words are related to ‘water’ and have a higher co-occurrence with their natural state than the opposite. When measuring the ratio between ‘ice’ and ‘steam’, a ratio close to 1 will depict a strong relationship between those words. In the example, this will be for ‘ice’ and ‘steam’ on water.

Unlike the very different approaches both Word2Vec and GloVe have, the performance is similar. The authors from GloVe claim that they can train faster and reach higher accuracy on a word analogy task, but without tuning the parameters of the Word2vVec model (Pennington et al., 2014).

2.2.4 ELMo

Deep contextualized word presentations are worded first by M. Peters et al. (2018). The researchers from AllenNLP use a deep bidirectional language model (biLM) to learn word vectors, opposed to the RNNs used in Word2Vec and FastText (Bojanowski et al., 2017; Mikolov, Sutskever, et al., 2013). This means that a left-to-right and a right-to-left LSTMs representations are concatenated. ELMo uses sentences as input as opposed to phrases, words or characters in earlier papers (M. Peters et al., 2018). The word representations are then computed on a two-layer biLM, altered by Jozefowicz et al. (2016) to have the most potential for a language model on large corpora and vocabulary sizes. M. Peters et al. (2018) argue that after pre-training the model, the weights of the model can be altered for a specific downstream task to perform best. The researchers found by exposing all the internal layers of the biLM, the word representations are deep.

M. Peters et al. (2018) give the example that the higher-level LSTM captures the context-dependent aspects of the words, while the lower levels have more syntactic qualities. ELMo representations are found to improve well on the SOTA (at that time) in all the different downstream NLP tasks and easy to integrate in NLP pipelines (M. Peters et al., 2018). Furthermore, ELMo shows domain transfer, learning domain-specific information by fine tuning the biLM on domain specific data that leads to increased downstream task performance (M. Peters et al., 2018).

2.2.5 ULMFiT

Universal Language Model Fine-tuning, or ULMFiT for short, is a transfer learning method to effectively train models for a wide array of NLP tasks (Howard & Ruder, 2018). The model uses a LSTM without additional features, but distinguishes itself on the fine-tuning method. Howard and Ruder (2018) claim to “significantly outperform” SOTA models and need only a fraction of labelled examples to effectively fine tune on downstream tasks.

Howard and Ruder (2018) took inspiration of the field of Computer Vision to create a transfer learning method aimed at NLP tasks. The ULMFiT method consists out of three stages, namely the language model pre-training, the language model fine-tuning and the classifier fine-tuning. The first stage is training the language model on a general domain, large dataset to ingest the properties of language (Howard & Ruder, 2018). The second stage is the fine tuning of the language model to the target task. Taking the target dataset, the language model is trained on this to gain a better understanding the task at hand. The last stage is the classifier fine-tuning. Here the pre-trained and fine-tuned language model is used to get a distribution of output probabilities to finally classify the input (Howard & Ruder, 2018).

The results of the experiments show that the model outperforms the SOTA significantly for different corpora and shows for supervised learning that a fraction of labelled examples are needed to match performance (Howard & Ruder, 2018). Furthermore, Howard and Ruder (2018) show that pre-training helps more on smaller to medium size datasets, ULMFiT fine-tuning works best on large datasets and that the ULMFiT classifier is the only classifier that shows excellent performance on all datasets.

2.2.6 Flair

German based research from the company of Zalando have had an character-level approach for a RNN model (Akbik et al., 2018). Where other models focus their LM on word levels, while still using character-level features nevertheless, contextual string embeddings are formed by only learning to predict the next character based on the previous characters. The Flair model's properties are that firstly, the model is trained without the notion of what a word is and secondly, given a context by the surrounding text, therefore having a different embedding based on when in a different context (Akbik et al., 2018).

This results in a model that is able to train on large, unlabelled corpora and shows to learn word meaning in context and produces different embeddings for different context. An upside from a character level approach is the ability to better handle misspelled words as well as gain an understanding of sub-word structures like pre- and suffixes (a big deal in the German language, home to Zalando) (Akbik et al., 2018).

Finally, this model (at the time) performed very well on sequence labelling and NER, posting the SOTA scores on these benchmark tasks (Akbik et al., 2018). The difference between character-, word- and sentence level model performance is not clear and therefore still cannot be concluded what model is superior. Figure 2.5 shows the character language model in combination with a sequence labelling model, in order to create NER tags based on concatenated character representations of words.

2.2.7 GPT

Generative Pre-Training (GPT) is developed by Radford and Salimans (2018) from OpenAI. The goal was to develop a model that is a combination of unsupervised pre-training and supervised fine-tuning, while the model contains a universal representation with little adaptation for various downstream tasks. The model uses

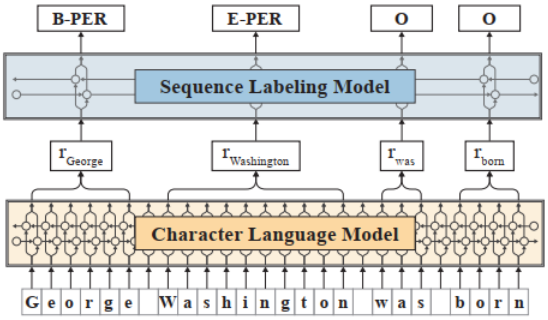


Figure 2.5: Flair neural network architecture (Akbik et al., 2018)

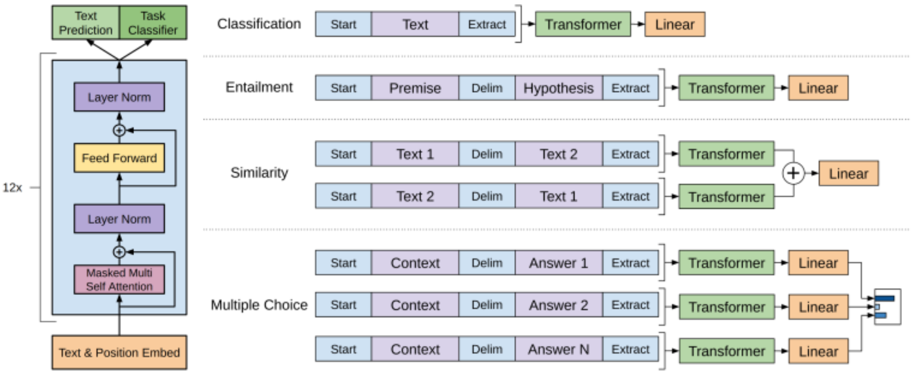


Figure 2.6: GPT showing the multitude of tasks it can be fine-tuned on (Radford & Salimans, 2018)

a left to right transformer that trains on a large text corpus. After training, the pre-trained parameters are adapted to the downstream task. The labelled inputs are passed through the pre-trained model and consequently the parameters are fine-tuned to the task at hand. Input can vary between task, where figure 2.6 displays the differences between different inputs. Multiple analyses show that the fine-tuning of the model increases accuracy and performance of the model.

2.2.8 GPT-2

GPT-2 is an extension on the model named above GPT by Radford et al. (2018). The researchers argue that language modelling has found its way to perform well on a specific task, but do not generalize well. Therefore, GPT-2 is a demonstration of a model that performs well on downstream NLP tasks in a zero-shot setting. This means that after pre-training, the model parameters or model architecture is not altered. This model achieved SOTA on seven of eight language modelling task for zero-shot tasks. GPT-2 is more known for the arbitrarily withholding the models because the models deemed to be too close to human written text and therefore a threat to society. Finally, the models were still released[3].

[3] https://openai.com/blog/better-language-models/

2.2.9 BERT

Bidirectional Encoder Representations from Transformers (BERT) stands for and is developed by Google (Devlin et al., 2018). BERT has the same approach in a biLM to predict the word as M. Peters et al. (2018), however it uses a masked language modelling (MLM) training procedure through a transformer architecture. This procedure means that randomly, words are masked in the input where the objective is to predict the masked word. Devlin et al. (2018) argues that unlike left-to-right language models, the transformer architecture enables the representation to combine the left and right context to train the model through MLM. Combined, a next sentence prediction task is added to train text-pair representations.

The BERT framework consists out of two parts, the pre-training and the fine-tuning of the model. Pre-training is the training over unlabelled data and fine-tuning is using the pre-trained parameters where after the parameters are fine-tuned in a supervised manner after using labelled data from the downstream task (Devlin et al., 2018). Unlike ULMfit, there is almost no change to the model except from changing the input and outputs of the BERT model (Devlin et al., 2018; Howard & Ruder, 2018).

Figure 2.7 displays this as on the left the pre-training objective, where unlabelled sentence pairs are trained using MLM. The pink blocks depict the tokens including separators and CLS (masked) tokens. The yellow blocks show the embeddings, that live in the blue block, the neural network. The green blocks are the values for every embedding after being processed through the transformer.

Finally, the output layer differs for the pre-training and fine-tuning phase and every fine-tuning objective itself. For example, in figure 9 the outputs for MNLI, NER and SQuAD have different classifiers. Only this changes and the pre-training and model stays exactly the same.

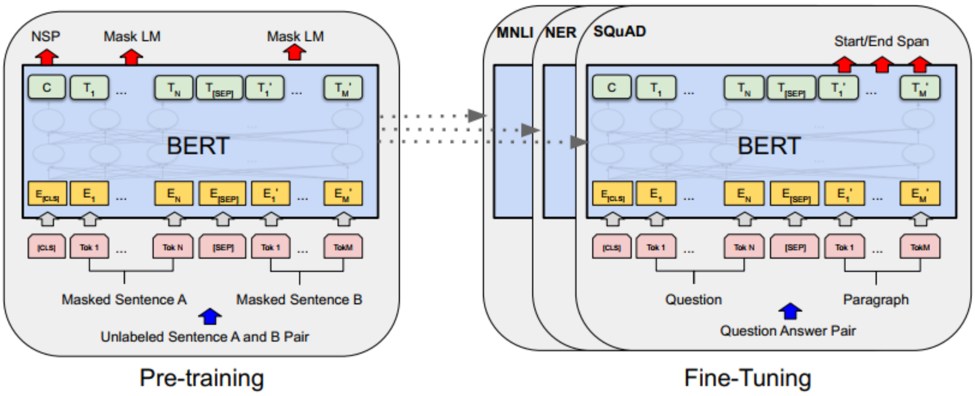


Figure 2.7: GPT showing the multitude of tasks it can be fine-tuned on (Radford & Salimans, 2018)

2.2.10 RoBERTa

A joint research from AllenNLP and Facebook have resulted in the paper and model RoBERTa (Liu et al., 2019). By running a replication study, the researchers found that BERT was "significantly undertrained and can match or exceed the performance of every model published after it” (Liu et al., 2019). Furthermore, the researchers have made some tweaks and changes to the training procedure and used more data to train their own model; RoBERTa. This results on SOTA results on GLUE benchmark and two other benchmarks.

2.2.11 XLnet

Where BERT achieved SOTA performance on using ‘real’ bidirectional context, the researchers from Carnegie Mellon University found a way to improve upon BERT with XLnet (Yang et al., 2019). By leveraging autoregressive (AR) language modelling, estimating the probability distribution of a text corpus with an AR model, XLnet succeeds in incorporating AR language modelling into a SOTA model competitive with BERT (Yang et al., 2019). An AR model is a feed forward model wherein the context of words earlier are used to predict the next word. GPT and GPT-2 are also AR models (Radford & Salimans, 2018; Radford et al., 2018).

Yang et al. (2019) argue that BERT suffers from the pretrain-finetune discrepancy. This is due to the MASK-symbols that are used in the MLM of BERT are missing in the fine-tuning task, therefore the pre-training is not representable for the fine-tuning phase. Additionally, BERT assumes that the predicted tokens are independent of each other, therefore simplifying natural language where there could be a dependency between two masked words (Yang et al., 2019).

Furthermore, there is a fixed range on dependencies where the range might be exceeded. XLnet uses Transformer-XL in their model architecture to solve this problem. Transformer-XL uses segment level-recurrent mechanism to capture the hidden state during training and reuses it on the next segment, therefore not losing long range dependencies. A constraint of XLNET is that an AR model only can feed the context one way. This means that the words in front of the to-be predicted word can be used as context, or backwards, where the words after the to-be predicted words are used as context.

2.2.12 T5

The T5 model, as of the end of October 2019, is the best performing model on the SuperGlue benchmark (A. Wang, Pruksachatkun, et al., 2019). T5 is a paper by Raffel et al. (2019) where a unified transfer learning approach is researched to gain a better understanding and push the current limits of the field. The definition of transfer learning given by Raffel et al. (2019) is the following; “Where a model is first pre-trained on a data-rich task before being fine-tuned on a downstream task.” The researchers argue that the advances of transfer learning are robust due to the massive amount of unsupervised training data that is available (20TB of data every month by common crawl), pre-training objectives, benchmarks and fine-tuning methods.

The Text-to-Text-Transfer-Transformer, or T5 for short, approaches an NLP task as a text-to-text problem. For the input text, there must be an output text. The text-to-text framework applies the same model, objective, procedure and decoding process to every task. This way, the effectiveness of different transfer learning objectives, datasets and other factors can be learned, while making progress to push the limits of transfer learning in the following ways: scaling up models and datasets (Raffel et al., 2019).

The model is based on the transformer architecture, where an encoder and decoder form the main components of the model (Vaswani et al., 2017). The model itself has minor changes in comparison to the transformer architecture. The model trains like BERT's MLM by corrupting 15% of the input tokens and let the model come up with the right one.

This paper is not only meant to create a SOTA framework, but is also interested in making a comparison of the current techniques, where the field stands and where it should go. Of course, the model is the biggest yet (11 billion parameters) and a new dataset is introduced (Colossal Clean Crawled Corpus).

2.3 Interpretation and visualization

Word representations are highly dimensional, for example a 300 (Mikolov, Sutskever, et al., 2013) to a 768 (Devlin et al., 2018) dimensional vector space. Analysis of word (and therefore interest) similarity will be difficult where dimensionality reduction can be a solution. principal component analysis (PCA) (Wold et al., 1987) and t- Distributed Stochastic Neighbor Embedding (t-SNE) (Van Der Maaten & Hinton, 2008) are two techniques that can help with interpretation and visualization of data through dimensionality reduction. More recently, the Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) method has made is its name through the use in the fields of cell biology, machine learning and social sciences. UMAP is comparable to t-SNE but converges quicker and better preserves global structure of the data, resulting in a better 'big picture' of the data. The most compelling argument to use UMAP is the implementation of cosine distance, a non-metric distance function especially designed for word vectors.

For the visualization of the data, Tensorboard[4] and Visdom[5] are widely used interactive visualization packages. Both are aimed at visualization tooling for machine learning experimentation. Both these packages seem an option to use for the data visualization of the model outputs, word representations and topic models from this thesis. A newer alternative would be streamlit[6], a python package to deliver interactive machine learning applications in a straightforward manner.

2.4 Conclusion

To conclude this chapter, from the perspective of different pre-trained models that are available and are or have been SOTA, the field of NLP has been taken various steps to better understand natural language and capture this in a pre-trained model. For this thesis, the question is how the models perform on the Dutch interest data to form vector representations and how well the models can 'understand' contemporary language of Dutch adolescents. The models provide the foundation for the analysis of the interest and therefore an integral part of this research. The better the representation of the Dutch language in the models, the better the performance on the downstream task will be.

[4] https://www.tensorflow.org/tensorboard

[5] https://ai.facebook.com/tools/visdom/

[6] https://streamlit.io/

Finally, another aspect is the usability and adaptability of the models and the available tools and workflows for incorporating the desired model into a NLP pipeline. This ranges from the adaptability of the model to the target domain to serving the results from the models to the end-user in a user-friendly interface.

Chapter 3

Method

This chapter presents the method used to choose, test and evaluate NLP models for the analysis of Dutch interest data. The rationale of the model's selection based on their unique properties is described in section 3.1. Secondly, The model baseline tasks for comparing the models on two downstream tasks is written in section 3.2. The model fine-tuning for leveraging transfer learning is described in section 3.3. Section 3.4 describes the qualitative method and the experimental setup for this integral part of the research. Lastly, section 3.5 is a description of the steps taken in creating the tool.

Where the different possible NLP models are explained theoretically in the last chapter, this chapter will focus on the usage of these models in a more practical setting. Researchers all over the world have been focusing on finding better methods to master NLP. To compare, standardized tests such as the Glue benchmark (A. Wang, Singh, et al., 2019) are used. A general remark for the current SOTA is 'the bigger, the better' (Raffel et al., 2019). Since the inception of transformer-based models (Vaswani et al., 2017), the basis does not evolve drastically, only the size of the number of parameters in the model grows (Sanh et al., 2019).

The downside to this, is testing the overall applicability of this model in real-world cases. While a model can work well on a test dataset for a 'standard' task, how does this translate to a more real-world experience? In order to find out, four models were chosen based on their unique properties in combination with their native Dutch training. The rationale for the Dutch training is the results of in the paper of de Vries et al. (2019), where the research shows it outperforms the multilingual trained BERT model (Devlin et al., 2018). The premise of being fully trained on a language instead of 104 different languages is also a logical choice. The different model's properties are important to link to the nature of the text, which in theory should make a difference in capturing the meaning in the model's representations.

In summary, the method will firstly consist of baseline tasks to create a common ground. Secondly, it moves to the quantitative similarity tasks in which the models will be trained and tested on. This will give insight in what model can perform best on a general NER and sentiment analysis. Thirdly, a fine-tuning step will be performed, for the models that allow, on web- scraped Reddit[1] data. The fine-tuned models will be tested on more qualitative tasks, to see which models performs best on more refined interest relation and embedding quality. In figure 3.1 below, an overview of the models, model adaptions and tasks are shown.

[1] https://reddit.com

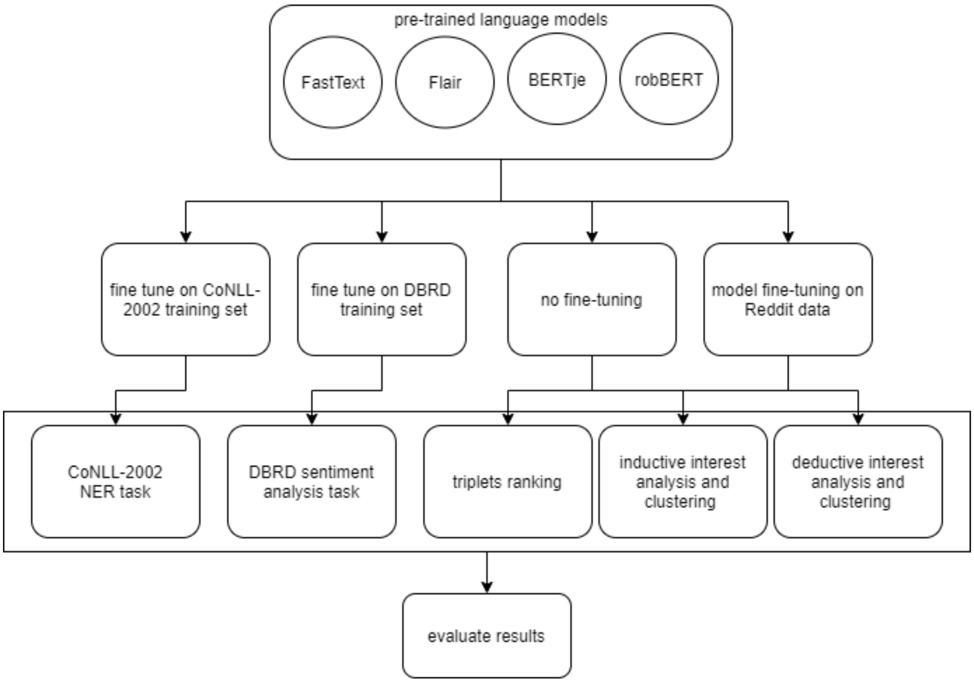

pre-trained language models
FastText
Flair
BERTje
robBERT
fine tune on CoNLL-2002 training set
fine tune on DBRD training set
no fine-tuning
model fine-tuning on Reddit data
CoNLL-2002 NER task
DBRD sentiment analysis task
triplets ranking
inductive interest analysis and clustering
deductive interest analysis and clustering
evaluate results

Figure 3.1: the model evaluation method

3.1 Model selection and rationale

The model selection is a process that takes into account not only the SOTA models, but also models that are expected to have good performance based on the target data. To tie this into our case, a short description of the data and data gathering process is described below.

The data gathering is done through an ESM method. This means that subjects fill out forms multiple times through the day, for 7 days in a row. In this case, this is every two hours, for a week. The subject answers three open questions; ‘Interest name’, ‘What was interesting about it?’ and ‘Why was it interesting?’. This results in event data with thousands of rows with different interests and their descriptions. The subjects are adolescents and the text fields are free, so there is no correction in language use and the fields contain colloquial Dutch.

The nature of the data is important for choosing the right model, since the model has to be able to handle the nature of the data well in order to perform. All the models chosen have their (dis)-advantages and provide a complete overview of different architectures and input types.

Due to the short nature of the data, interests are described most of the time in only a few words. The case could be made that sentence encoders (transformer-based architectures such as BERT and RoBERTa) will not work well. However, using the extra text fields that give a rationale for the interest, the combination of sentences would be ideal for the sentence-based models. Word embeddings for word-based data sound like a good combination. However, while a five year old method sounds new, in the field of NLP this is relatively old. Because these word embeddings do not take into account the context, interests could very well be represented poorly. A character-based model could be a solution to misspellings and underrepresented word forms, due to the architecture of the model that uses the surrounding characters as context. The big question is; can a character-based model without the notion of words and sentences understand interests?

Taken into account the aforementioned variables, four models are selected for the comparison. These models are chosen since they are pre-trained on Dutch corpora and differ in the input and architecture. The three methods all have their pros and cons, have stood the test of time or are the current SOTA on many downstream tasks.

First off, the selection off FastText (Bojanowski et al., 2017) will give an idea what the performance is of the 'traditional' pre-trained word embeddings that has lighted the NLP fire back in the beginning of the 2010's. The OOV functionality of the FastText embeddings are able to handle misspellings and variations that will hopefully yield good results on the target domain.

Secondly, the character-based model Flair (Akbik et al., 2018) has shown SOTA performance on the NER task while other institutions were focusing on making whole sentence encodings work. With the notion of a contextualized, character-based model, it completely differs from word- and sentence models while maintaining the ability to capture linguistic concepts such as words, sentences and even sentiment (Akbik et al., 2018).

Finally, transformers have changed the architecture of pre-trained models indefinitely and show this will yield the most performance-wise (Vaswani et al., 2017). Therefore, the inclusion of BERT (Devlin et al., 2018), the first-mover of all the sentence-based pre-trained models, is an inclusion that would be self-evident. Finally the addition of the RoBERTa model, which is also uses transformers (just as BERT but differs on slightly which improves overall scores on downstream tasks (Liu et al., 2019). To expand on the four chosen models, the four models are described in more detail in combination with their respective implementation and framework used.

For the FastText model, the model will live in the Flair framework (Akbik et al., 2019). This framework gives easy access to loading different kinds of word embeddings in different languages. For the FastText model, a model created by FastText itself is used, trained on Dutch Common Crawl and Wikipedia. This results in embeddings of dimension 300.

Just as FastText, the character embeddings live in the Flair framework. The Flair embeddings, based on Flair, have been trained on Wikipedia and Dutch version of OPUS, a multilingual open corpus (Akbik et al., 2018). The flair embeddings are actually a combination of two models, namely a forward- and backward model. This results is combined and projected to a dimension of 512.

For the other two models another framework is used, namely HuggingFace's Transformers (Transformers) (Wolf et al., 2019). This library is a high-level API to all sorts of transformer based models. Users can share their own models. The Dutch implementation of BERT, called BERTje (de Vries et al., 2019), is initialized through Transformers. BERTje is a transformer-based, sentence-oriented model with a dimension embedding of 768. BERTje is trained on a composition of 5 different datasets, combining into 2.4 billion words and a size of twelve GB of text (de Vries et al., 2019).

The last model is RobBERT (Delobelle et al., 2020). RobBERT is based on the RoBERTa architecture and closely related to BERT. RobBERT has shown promising results in downstream tasks where it outperformed BERTje. The training procedure is slightly different and the data for pre-training differs. The dataset used for RobBERT is the Dutch section of the OSCAR corpus, consisting of 6.6 billion words and totaling in 39GB of text (Delobelle et al., 2020).

To summarise, table 3.1 contains the model names, accompanied with their input type, architecture and the corpus trained on.

Model name	Input type	Architecture	Trained on
FastText	Word-based	RNN	Wikipedia/Common Crawl
Flair	Character-based	Forward and backward RNN	OPUS
BERTje	Sentence-based	Transformer	Books, TwNC, SoNaR-500, Wikipedia, Web News
RobBERT	Sentence-based	Transformer	OSCAR

Table 3.1: An overview of the input type, architecture and pre-training datasets

3.2 Model baseline tasks

To create an overall assessment of the different NLP models, two baseline tasks are identified, trained on and evaluated. The first baseline task is the CoNLL 2002 NER task (Tjong Kim Sang, 2002) where the model has to recognise four entities. The entities are person, location, organization and miscellaneous. The dataset is provided through the Flair framework, including the code to run this task for both the Flair and FastText embeddings. The training runs for 50 epochs, a full cycle over all the training data, while the other hyperparameters are kept at standard. The transformer models are trained for the same amount of epochs in the Huggingface transformers framework.

The second task is a sentiment analysis binary classification task on the 110k Dutch Book Review Dataset (110kDBRD)[2]. The dataset consists out of 110 thousand book reviews scraped from Dutch book review website Hebban[3]. A balanced training subset of more than twenty thousand reviews is trained, followed by a ten% test set to evaluate the model. After training for four epochs, the model is tested on the test set and the MCC score is calculated. MCC is a widely used metric when using imbalanced data in different fields, originating from bioinformatics (Matthews, 1975). This metric has been used for downstream task evaluation for NLP models such as CoLA[4] (A. Wang, Singh, et al., 2019).

3.3 Model fine-tuning

To adapt a pre-trained model, language model fine-tuning methods are used in order to leverage transfer learning. Given a corpus, a model can be fine-tuned by using the training objective that is also trained with for the pre-training. Both Flair

[2] https://github.com/benjaminvdb/110kDBRD

[3] https://www.hebban.nl/

[4] https://nyu-mll.github.io/CoLA/

and Transformers have fine-tuning capabilities, FastText however, has not. For the language model fine-tuning, the presets have been used given by the example code of both frameworks.

For the corpus to train the model on, Reddit has been used as source. Reddit is a social news website where people can post links, text, photos, videos and more. Under these posts people can reply, debate and comment on the posts. Reddit is popular under a large audience, especially adolescents. Reddit consists out of subreddits, a kind of hierarchy of subjects. Because the aim is to gather Dutch text, the subreddit '/r/thenetherlands[5]' is used as source. The subjects on this subreddit are all related to the Netherlands and in Dutch. People post news articles to discuss, post pictures or ask questions.

To create the corpus, Python Reddit API Wrapper (PRAW)[6] has been used. The top 1000 posts have been selected and from these 1000 posts, all the comments have been collected to form the corpus. Minor edits have been done to the dataset such as removing hyperlinks, deleted and empty comments. This results in a dataset of over 70.000 comments.

3.4 Domain specific tasks

In order to test and compare the different models on the quality, three different experiments will be conducted.

1. A newly-designed test that will be used in order to assess the quality of the embeddings of the different models.
2. A retrospective, blind test that will be executed where over a larger set of interests where the distribution and clustering of interests will be analysed.
3. A test that measures the relationship between different sports, musical instruments and middle school classes.

For the first two tests, interest data from the ERC research is used in order to make sure that the results of the experiments reflect the same environmental variables as the models would face in practice. For the third test, data from Wikipedia categories is used.

3.4.1 Triplets analysis

The goal of the first experiment is to see if the model interprets interests in the same way experts do. For the first test, the method of the experiment is as follows. The data consists out of 100 samples. one sample consists out of three interests, forming a triplet. The three interests are embedded by the model and returned. To see if the model understands the different interests, the relative similarity of the embeddings is calculated, using cosine similarity[7]. The closer the cosine similarity score is to one, the more similar the interest are. For example, the three interests in the hypothetical triplet are football, golf and watching tv. Both football and golf are

[5] https://www.reddit.com/r/thenetherlands/

[6] https://praw.readthedocs.io/en/latest/

a sport, so those two interests will be relatively closer together than the similarity score that football-watching tv and golf-watching tv. Arguing that football is on tv more often than golf and is watched by a larger audience, the similarity of football and watching tv is relatively higher than golf and watching tv. We got a classification of these three interests now. Table 3.2 shows a representation of how the similarity scores are depicted from the hypothetical example above.

Interest	football	golf	watching tv
football	1	0.75	0.5
golf	0.75	1	0.25
watching tv	0.5	0.25	1

Table 3.2: Interest similarity matrix

This classification is done for the 100 triplets. The example from above is relatively simple to do, but if the randomly chosen interests are not evidently similar, it can become quite hard to make a classification of relative similarity. Therefore, a structured process is created in order to classify the interests. The experts have created a 7-dimensional interest relation classification form to score the relation between interests on different aspects. The triplets are created by the experts separately and later on discussed until agreement of the scores. The full form is in appendix B. The interest's relations are evaluated on the aspects in table 3.3 below.

Name	Explanation
Time: rhythm and regularity	When the interests are time bound, are these comparable to the notion of regularity?
Specificity of knowledge and skills	When there is knowledge and skill needed, are these comparable?
Societal knowledge of phenomenon	When societally known, are the interests the same in the matter of culture and history? Both on a micro- (inside the two interests) and macro level (for a broader audience that share culture and history)
Material comparability	When materials are needed to practice the interest, are these materials comparable?
Geographical comparability	When bound to a physical or digital space, are these comparable?
Social necessity and social nature	When bound to someone else, how comparable are these people?
Link to institutions	When bound to institutions, how comparable are these?

Table 3.3: The dimensions used by the experts for assessing the relation between two interests

Finally, the classification of the models is compared to the classification of the experts. To measure the ranking of the relative interests, there are three groups. Interest classification done totally correct, so all interests correlations are ranked

[7] https://scikit-learn.org/stable/modules/metrics.html#cosine-similarity

correctly. Secondly, ranking where only one interest correlation ranking is correct of the three. Finally there are the rankings that are totally incorrect. To inspect further, a set of triplets in the expert scores are chosen. These triplets are compared on ranking, but also on the correlation scores to see if there are notable differences between expert and model evaluation, but also between fine-tuned and non-fine-tuned models.

3.4.2 Inductive interest analysis and clustering

To move to the second test, the goal changes to a more broader approach to interest analysis and how the interests relate. Where the first test looked and individual samples where three interests were inspected closely, this test looks at the big picture where the structure of hundreds of interests are investigated.

The second test is a blind test where experts get a two-dimensional plot of all the unique interests that occur in the data gathering period for one school. The interests are not altered, only embedded by the different models. In order to bring back the highly-dimensional data that is returned from the models to a two-dimensional plot, UMAP is used (McInnes et al., 2018). The plots are filled with annotated data points. The plots are interactive, meaning there is the option to zoom in/ out, making sure the researchers can inspect the plot thoroughly. Furthermore, the researchers are also provided with an unsupervised clustered set of interest, using HBDSCAN (McInnes et al., 2017).

While investigating the plots, the researchers are given the task to take into account the distribution of the interests, looking for a structure of related interests grouped together, recognising overarching practices or the odd one out. Using the different plots and comparing the different structuring, the desire is to better understand the semantic connections the models make.

Additionally, the researchers are also provided a spreadsheet with a list of the different clusters and outliers (appendix C). The researchers have to name the clusters and see if the clusters have interests that do not belong to the cluster. Furthermore, the experts describe what they think the factor is that the cluster is based on. Lastly, the list of outliers is inspected in order to see if the outliers do not fit into one of the clusters. The naming of the clusters is conducted in order to indicate the coherence of the clusters and the possible outliers that can adhere to this cluster.

3.4.3 Deductive interest analysis

The third and last test is investigation of the ability to categorize words correctly. For this categorisation, the Dutch Wikipedia category tree[8]is used. Different categories are sought out and embedded. The hypothesis of this test is the following: Given the category-names and children of the Wikipedia category tree, the children of the same category should be correlated together. The model is given a set of middle school courses, instruments and sports. The categorisation is done by only one set at a time. The spreadsheets used for this analysis are found in appendix F.

To illustrate, the categories are countries and the children are cities. If we have the Netherlands and Germany as categories and Amsterdam, Utrecht, Maastricht, Dusseldorf, Berlin and Hamburg as children, one would expect that the Dutch city data points will be closer to the Netherlands data point than to the Germany data point. Same for the German city data points to the Germany data point.

3.5 Tool

3.5.1 From data to embeddings

For converting the text to embeddings, a python notebook is written to embed the sentences to vectors. As a rough outline, the text, the tokenizer and model are loaded. Then, one for one, the text will be tokenized, used as input for the model and the output of the model will be saved to a list. This list will be converted to a file in order to be used by the visualisation and insight process.

3.5.2 From embeddings to insights

For the visualisation and interpretation of the embeddings, Streamlit is used to provide an interface. The outline is simple. The user can upload the embedding file created in the step above. Streamlit will give an overview of the loaded interests and the user can select different analysis methods to gain insight in the relations between the embeddings. The user can summon a two-dimensional plot using UMAP (McInnes et al., 2018) as dimension reduction technique, a correlation matrix and a heatmap from the correlation scores. The correlation matrix can be saved as a .csv file in order to further analyse the data. Screenshots of the tool are provided in appendix E

[8] https://nl.wikipedia.org/wiki/Categorie:Alles

Chapter 4

Execution and results

This chapter will describe the execution of the 5 different tests and subsequently walk through the results from these respective tests. The tests are divided into the quantitative (section 4.1) and qualitative tests (section 4.2). This yields a dense chapter with a lot of information. In order to keep the story unscrambled by code and important but cluttering work, the code is moved to the code repository[1]. For each test, the spreadsheets, plots and other figures are moved to the appendix (B,C,F) that is worded in the section. The interpretation of the results is described in section 4.3 and conclusions are made in section 4.4.

4.1 Quantitative tests

The quantitative part of the tests consists out of two standardized tasks in order to create a baseline to evaluate the models on. The tasks are described in more detail in section 3.2. The baseline tasks are only evaluated with the standard, non-fine-tuned models, since the effect on the model from the fine-tuning will only be indicated by the qualitative tests where domain data is used. Section 4.1.1 describes the CoNLL 2002 NER task and section 4.1.2 describes the execution and results of the Dutch Book Review Database (DBRD) sentiment analysis task.

4.1.1 CoNLL 2002 NER task

To state the precise implementation, the models have been trained on the training set for 50 epochs and scored on the test set. The code of this training task is inspired on the code provided by the Flair framework[2]. The scores are expressed in F1-score, accompanied with precision, recall and the training loss of the model. The scores are stated in table 4.1, the order based on the F1-score from best to worst.

The F1-score has a scale from 0 to 1 and consists out of the harmonic mean of precision and recall. The F1-scores all fall between a range of 0.1 difference. For the precision, this window is only 0.01858. Recall has more spread between low .90 and high .78. The lowest training loss is recorded at 0.0881 and the highest training loss is 0.7698.

[1] https://git.science.uu.nl/tvdermeer/thesis

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

Model name	F1-score	Precision	Recall	Training loss
BERTje	0.90309	0.89961	0.90660	0.0881
Flair	0.87603	0.88088	0.87195	0.1317
RobBERT	0.86372	0.86216	0.86528	0.1091
FastText	0.82795	0.88103	0.78928	0.7698

Table 4.1: Results from the CoNLL 2002 NER task

4.1.2 DBRD sentiment analysis task

For this task, the models have been trained for four epochs on the training set before being validated on the test set. The execution of the fine-tuning and evaluation is based on the different sources.[3][4][5] The scores of the sentiment analysis task are shown in table 4.2.

The MCC metric has a scale of -1 to 1. A score of 1 is a perfect score, so all the scores are equal to the test set truth. A score of 0 is equal to random chance. A score of -1 is total disagreement.

Model name	MCC-score
BERTje	0.85072
RobBERT	0.76551
FastText	0.60433
Flair	0.54329

Table 4.2: Results from the DBRD sentiment analysis task

4.2 Qualitative tests

4.2.1 Triplets analysis

Firstly, the interests from the triplets are embedded. This is done with the help of a GPU on Google Colab[6]. General purpose packages such as Pandas and Numpy are used for this process. The interests are put in a list, the model is loaded and one for one embeds the interests. For the model parameters please see section 3.1. Please note that the transformer-based models (BERTje, RobBERT and fine-tuned models), are loaded through the package Transformers. For the Flair, fine-tuned Flair and FastText model, the Flair framework is used. The embedded interests are converted to a .csv file, in order to be processed by the next step.

For the test, the triplets need to be rank on the inner similarities between the triplets. Therefore the similarity score between interest [1,2] [1,3] and [2,3] are computed using the scikit-learn cosine similarity metric. Now that pairs have a similarity score, the pairs are ranked from most similar to the least similar pair. These ranks are compared against the expert evaluations.

[3] https://github.com/flairNLP /flair/blob/master/resources/docs/TUTORIAL_7_TRAINING_A_MODEL.md

[4] https://github.com/huggingface/transformers/tree/master/examples/text-classification

[5] https://github.com/iPieter/RobBERT/blob/master/notebooks/finetune_dbrd.ipynb

[6] https://colab.research.google.com/

Model name	Correct	Partially correct	Incorrect
RobBERT (FT)	29	39	32
FastText	27	47	26
Flair (FT)	24	45	31
Flair	23	43	34
RobBERT	23	43	34
BERTje (FT)	20	46	34
BERTje	17	54	29

Table 4.3: Results from the triplet ranking test. Note that the fine-tuned models have the FT in the model name

Table 4.3 shows the results of the different models. The columns show the number of correct, partially correct and incorrect rankings compared to the expert evaluation. The models followed by the letter FT are the fine-tuned models. The RobBERT model that is fine-tuned on Reddit texts is performing the best with 29 of the 100 correct rankings. FastText is second, while have less rankings correct it has more partially correct answers and less incorrect answers. Flair fine-tuned, Flair and RobBERT are all close together, with Flair and RobBERT scoring the exact same score. Both the BERTje models have the lowest number of correct scores of all the models.

4.2.2 Inductive interest analysis and clustering

For this test, the unique interests of one school are used. The number of interests count up to 455. The unique interests are embedded just as in the first test. The list with embeddings has to be reduced to a two-dimensional point to be plotted. For this, the dimension reduction algorithm UMAP (McInnes et al., 2018) is used. The metric used is ‘correlation’ and a transform seed of 42. Furthermore, the parameters are left as-is. The interactive plots are made using Bokeh.

For the clustering of the interests, HDBSCAN (McInnes et al., 2017) is used. However, dimension reduction is first done with UMAP to 50 components, thereafter HDBSCAN handles the last 50 to two dimensions. This is due to the best practices described in the documentation of HDBSCAN’s package[7].

For the evaluation method, the experts are given a spreadsheet which contains the clusters and outliers. The clusters are named based on the content, given comments on the good and the bad of the cluster and a score on how coherent the cluster is. The evaluation is done separately by two experts. Their evaluation is compared and contrasted. The extensive results from all the spreadsheets are in appendix C.

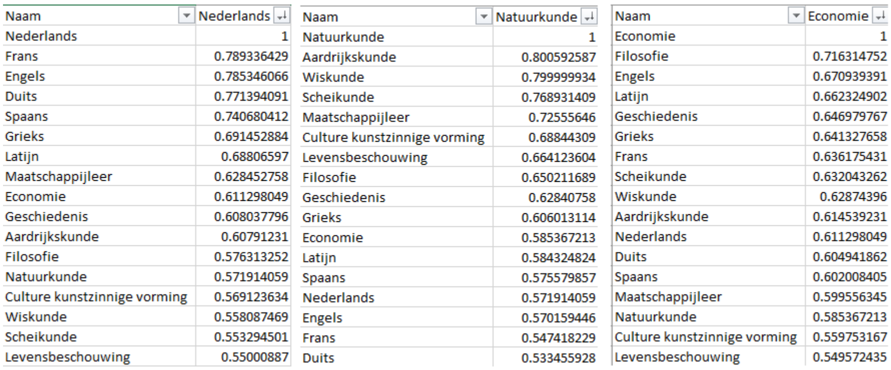
4.2.3 Deductive interest analysis

For the last analysis, three categories are created. Sports, instruments and middle school courses. The sports and music instruments are retrieved from Wikipedia categories. For the music instruments, the names under the Hornbostel-sachs scheme are used with the names of instruments as children. In order to bring back the

[7] https://hdbscan.readthedocs.io/en/latest/faq.html#q-i-am-not-getting-the-claimed-performance-why-not

number of sports and instruments, a selection was made to make sure the items are correct and more generally known, so that the item is probably known in the vocabulary of the model too. The middle school courses are broken down into alpha, beta and gamma courses.

The models receive the lists of sports, instruments and courses as input and embed those into the vector representations. The correlation score of these vectors are computed and a matrix is created to have a comparison to all. The researchers take a sample that is consistent over all models to check how the instruments and sports are correlated. For the school courses, all items are checked.



Naam	Nederlands	Naam	Natuurkunde	Naam	Economie
Nederlands	1	Natuurkunde	1	Economie	1
Frans	0.789336429	Aardrijkskunde	0.800592587	Filosofie	0.716314752
Engels	0.785346066	Wiskunde	0.799999934	Engels	0.670939391
Duits	0.771394091	Scheikunde	0.768931409	Latijn	0.662324902
Spaans	0.740680412	Maatschappijleer	0.72555646	Geschiedenis	0.646979767
Grieks	0.691452884	Culture kunstzinnige vorming	0.68844309	Grieks	0.641327658
Latijn	0.68806597	Levensbeschouwing	0.664123604	Frans	0.636175431
Maatschappijleer	0.628452758	Filosofie	0.650211689	Scheikunde	0.632043262
Economie	0.611298049	Geschiedenis	0.62840758	Wiskunde	0.62874396
Geschiedenis	0.608037796	Grieks	0.606013114	Aardrijkskunde	0.614539231
Aardrijkskunde	0.60791231	Economie	0.585367213	Nederlands	0.611298049
Filosofie	0.576313252	Latijn	0.584324824	Duits	0.604941862
Natuurkunde	0.571914059	Spaans	0.575579857	Spaans	0.602008405
Culture kunstzinnige vorming	0.569123634	Nederlands	0.571914059	Maatschappijleer	0.599556345
Wiskunde	0.558087469	Engels	0.570159446	Natuurkunde	0.585367213
Scheikunde	0.553294501	Frans	0.547418229	Culture kunstzinnige vorming	0.559753167
Levensbeschouwing	0.55000887	Duits	0.533455928	Levensbeschouwing	0.549572435

Figure 4.1: three examples of the correlation scores on middle school courses

4.3 interpretations

This subsection is expanding upon the results above. By giving the results more context, diving deeper into the different analyses and finally make an interpretation of the results, the goal is a meaningful outcome of the executed work. Section 4.3.1 will expand on the interpretations for the quantitative tests. This is divided into the CoNLL (section 4.3.1) and DBRD (section 4.3.1) tasks. This is followed by the qualitative tests, described in section 4.3.2. Again, this is subdivided into the triplets, the inductive interest analysis and clustering and lastly, deductive analysis. The conclusions for this chapter are provided in section 4.3.

4.3.1 Quantitative tests

The quantitative tests serve their purpose through comparing the models on a task that reflects a real-world approach to text analysis and use of real-world data. Specialized test suites have been developed in order to thoroughly compare and contrast models on their performance on a wide range of downstream tasks (A. Wang, Pruksachatkun, et al., 2019; A. Wang, Singh, et al., 2019). Unfortunately, these are ended on the English language and not available in Dutch. On the other hand, there are Dutch tasks available but fewer (Tjong Kim Sang, 2002) and older, not testing the full capabilities of LM.

CoNLL 2002 NER task

The models are close in performance on this task, especially in the precision metric, where the models score between .03 from each other. The difference is made in the recall of the models, where the FastText is scoring considerably lower than the other models. RobBERT scores well in both categories but just a little less than the Flair model does. On top is the BERTje model with a .9003 score.

Another metric that stands out is the training loss. The training loss is the error the function has on the training set while training. The lower the training score, the better the model can fulfil the task on the training set. If the model can generalize well and does not overfit on the train set, this will result in a high score on the test set. The FastText model seems to not fit to the test set well and it shows in the training loss (0.7698). It does not come close to the training losses of the other 3 models. What stands out is that with a higher training loss, 0.1317 to 0.1091, Flair gets to outperform RobBERT on the test set. It cannot be concluded that there is overfitting but it can be that RobBERT does not generalize well from the training- to test set.

DBRD sentiment analysis task

The DBRD sentiment analysis task consists out of sentences that are either positive (1) or negative (0). Just as the NER task, the models are trained, however for a significant number of less epochs (50 on NER, four on sentiment analysis). This is due to the recommendations of the tests setups, where the NER task advices a maximum of 150 with a patience of 5. Patience means that after five epochs of no improvement, the model after the epoch with the lowest training loss is used. The sentiment analysis task only advices four epochs. While training the models on the sentiment analysis task, training took up to 8 hours to train for the Flair model.

Unfortunately, the output of the models for this task is very brief, only returning the MCC score. Flair and FastText do have the hardest time to perform in this task, both scoring considerably lower than the Transformer-based models. The big difference is the nature of the data, where the task consists of understanding the sentiment of full sentences. If there is an interplay between words, one could argue that transformer-based models have an easier time understanding through their attention mechanisms than through a RNN for both the other models. As a general remark, the difference in training method can have an influence on the final scores.

4.3.2 Qualitative tests

The qualitative tests have much more room for interpretation. These tests focus on the quality of the embedding layer of the models and therefore do not return a score, result or metric. The embedding layer returns a vector representation of the input, a translation so to say from text to computer-interpretable numbers that can be used to do computations on. The metric that is used often by the qualitative tests, is the cosine similarity. A measure that returns a result between 0 and 1 where 1 is complete similarity, two inputs that are precisely the same, and 0 no similarity.

Triplets analysis

These aspects are ranked from a one (no) to 5 (very much). The average of these aspects result in a score between one and five that shows the strength of the correlation between the interests. The expert evaluation method creates a right skewed distribution of interest correlations. The correlation score distribution of the models on the hand, is closer to a normal distribution.

Also important to note, the expert evaluation ranking consists of 44 ties. This means that two of the three scores are ranked equal and therefore the ranking does not consist out of [1,2,3] but out of [3,1,1] or [1,3,,3]. The likelihood of the models returning two equal correlation scores is very small and therefore it is safe to say that there are no equal ranks in the ranking of the models. Therefore, for 44 cases, the correct prediction cannot be made. The models that return two of the three interest pairs correct in a triple, that triple is deemed correct.

A comparison of every correlation of every triplet could be done in order to find out more about the precise nature of the embeddings. This is deemed unfeasible due to the amount of work that would be created in combination with the other tests. Therefore the choice is made for a selection of triplets that hold distinctive properties to see how the models cope with those properties. To dive intp more detail, six triplets are chosen that are distinctly different and hold a relation in for example a social activity, a school related subject or a individual activity. The activities are numbered as in the triplets set, and are shown in table 4.4. Again, the full triplets analysis can be found in appendix B. For every model, the triplets are discussed.

To start with the FastText model, it has made only one mistake (No. 80) from the selected triplets, while all the other models had at least two full mistakes. As for this model and almost all the other models, it has 'Nieuws' correlating high with 'Drinken met vrienden', while to the experts it is given that 'Drinken met vrienden' and 'Dingen doen met vriendinnen' is strongly related (a score of 4.29 by the experts, [1,5]). For FastText, this correlation was strikingly low.

For the RobBERT model, there were two mistakes in the triplets. As mentioned above, no. 80 was wrong but also triplet No. 5. The experts ranked the 'Video editen' and 'Creatief bezig zijn ...' as the highest ranked interest relation. This is not shared by the model, that ranks this relation the lowest. The relation of 'Video editen' and 'Zingen' is strong by most of the models, except for the FastText model.

The comparision of the RobBERT model to its fine-tuned counterpart is interesting. The idea behind the fine-tuning is that the model would be better in handling the interest data. When inspecting the six triplets chosen for this analysis, the model got a better score on the number of totally correct triplet rankings while the number of incorrect stays roughly the same (increase of six more correct answers and two incorrect answers less). The models both have more or less the same incorrect answers, but the answers that RobBERT has partially correct, are done totally correct more often by the fine-tuned RobBERT model. A notable difference when looking at the correlation scores distribution of the fine-tuned and non-fine-tuned model is that the fine-tuned model has a greater spread. This is not the case for the BERT model when it is fine-tuned.

To go into more detail for the BERTje model, the triplet of interest is No. 56. The experts ranked this triplet with relatively big difference between the interest sets. To be more precise, the relation between 'Bio-Informatica' and 'het vak geschiedenis en scheikunde' is the strongest (3.57), followed by 'het vak geschiedenis en scheikunde' and 'Lezen zowel literatuur als manga' (2). The relation between 'Bio-Informatica' and 'Lezen zowel literatuur als manga' is the lowest (1.29). When looking at these scores, the decision is clear cut. However, when looking at the scores of BERTje, and also for the fine-tuned version, the scores are relatively close to each other (between a range of 0.1). Also, the relation between 'Bio-informatica' and 'het vak geschiedenis en scheikunde' is not curated as high as the experts do. For both the models, the relation as both beta disciplines are not recognised.

To compare the fine-tuned model of BERTje to the standard BERTje model, the BERTje fine-tuned model got more answers correct (three extra), but also more answers incorrect (five extra). The correct and incorrect triplets are roughly the same for both the models. Finally, the distribution difference as seen by the RobBERT models, is not occurring for this model. Where there is an clear improvement for the robBERT model by fine-tuning, this is not as clear for the BERTje model.

The models that are left, are the Flair model and its fine-tuned brother. The scores get better by fine-tuning, but only minimally (one extra correct, three less incorrect). However, when looking at the incorrect scores for both the models, only twelve of the 31 are similar. This means that roughly twenty incorrect answers do not overlap. This is the same for the correct answers, where only ten are of the 23 are the same. This phenomenon can be the product of the fine-tuning heavily affecting the model where there is previous information lost in the model.

No.	Interest 1	Interest 2	Interest 3
3	Films kijken op netflix	Serie kijken	Toneelspelen
5	Video editen	Zingen	Creatief bezig zijn, (...) schilderen en handlettering
17	Met vrienden chillen	Basketball	Sociaal doen met vrienden
23	Informatica	School	Sporten
56	Bio-informatica	Het vak geschiedenis en scheikunde	Lezen zowel literatuur als manga
80	Nieuws	Drinken met vrienden	Dingen doen met vriendinnen

Table 4.4: The interests used for deeper analysis

Lastly, looking at the distribution of the correlation scores, the range of scores is roughly the same for both the models. However, the models differ on where the range is between 0 and 1. The Flair base model has a range between 0.16 and 0.44, while the fine-tuned model has a range between 0.31 and 0.66. To compare this to the other models, the correlation scores are typically between the 0.4 and 0.8. When contrasting the Flair models, the embeddings are effectively getting more similar, changing the distribution to higher scores.

Inductive interest analysis and clustering

The general conclusion of the experts was that the results were not as expected. The experts evaluated the clusters as superficial and morphological, not given the substantive meaning of the embeddings. Clusters are made from similarities in verb/noun, abbreviation and matching words, not the meaning.

The inductive interest analysis and clustering focuses on more of a production setting where the simulation of the use of visualization and clustering methods is applied. The main driver for using this method is the automated clustering that takes away the difficult task of creating coherent clusters from all the interests. The researchers found that doing this by hand, the number and coherence of clusters could easily change when new interests were introduced. Secondly, the visualization of interests helps to grasp the distribution of interests and how they relate to each other.

The analysis of the models by using the spreadsheets has taken away the view on the whole plot and therefore the general distribution of interests in a two-dimensional space. On the other hand, the clustering is as important, because interests that are in the cluster but do not belong, are also close to the interest in the cluster in the plot.

Another reason in favour for using the spreadsheet over the plot, is the degree of manageability that the spreadsheet brings. The only focus is the coherence of clusters, not a full plot with tens of clusters and the coherence therein. When the clusters are right, the coherence between is a viable option. However, the level of correctness was not reached by any of the models.

The models were all suffering from the same shortcomings. Firstly, the degree of how the models react on parts of words that are equal, for example the use of social-. To extend the example, Social media, Social worker are two very different things. Secondly, the models seem to react on characters that do not often occur. For example, the Flair FT model creates a cluster that is probably linked with each other because it contains brackets. The RobBERT FT model looks like it makes a cluster because it are abbreviations. For every model, multiple examples can be pointed out.

The researchers have opted to not continue with filling out the outlier to which clusters, since the coherence of clusters was not on a level to say with certainty if the outlier would fit the cluster. This results in a test that has unfortunately failed by all the models, and is not ready to be used in production.

Ultimately, the question was raised if the plotting and clustering methods do influence the embeddings so much that the interpretation of the embedding is not 'pure' anymore. To expand on this, do the dimension reduction technique in UMAP and the clustering algorithm HDBSCAN take away the subtleties from the vector representations? Closer inspection of the similarity matrices made for the models to inspect discrepancies between the models and the plots, shows that there was no such thing found. The resulting two-dimensions from the vectors of the embedding layer was not effected significantly by the dimension reduction techniques and the clustering method was seemingly working the way it should.

To investigate the raw embeddings more, the deductive interest analysis takes a look at the similarity scores between items that are widely understood as similar in a sense and therefore can verify the quality of the embeddings. The underlying problem here is the degree of meaning the model has of the language. The interpretation of the researchers is that the model focuses on a lot of the morphology of words, but not necessary in semantics of words.

Deductive interest analysis

For the final test, the interest analysis is executed using hierarchical structures from Wikipedia and analyse the embeddings on similarity between equally categorised items. The number of items for sports and instruments was too big to analyse all.

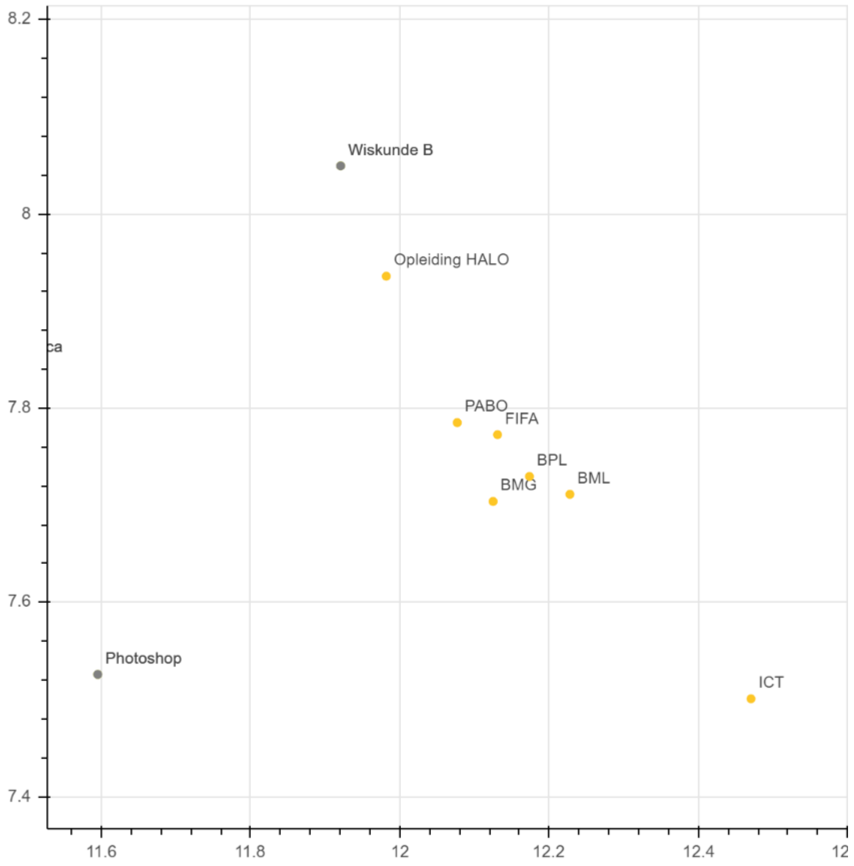


Figure 4.2: a cluster based on abbreviations

Therefore the same sample is taken for every model. The middle school courses are all analysed, since this number was more manageable.

The middle school courses, broken down in alpha, beta and gamma, worked generally well for all the models. Most of the models could distinguish the languages very well, except for flair FT that had difficulties with Latin. BERT could not correlate French high with the other languages and FastText could not work well with German. The beta courses also share the same word part in '-kunde'. Therefore this score is less significant because the high correlation can also be based on consisting of the same word part. For the gamma courses, this was the hardest of the middle school course categories. Multiple models had difficulties to distinguish the courses like biology, geography and economy from the beta courses. Math was scoring high on the gamma courses for BERTje. For RobBERT, history was correlating the highest with math. For multiple models, Philosophy was correlating high with beta courses.

Unfortunately, the good performance of the models on the courses is not continued in the sports categories. The sample consists of American football, badminton, bridge, hockey and bergsport. For American football, the models could not understand well that it was a ball sport. The Flair model was correlating English words high with American football. BERTje correlated words with 'bal' in it. For badminton, a racket sport, tennis was nowhere to be found in all the models. FastText correlated 'rolstoel' high with badminton. FastText was working relatively well on the other sports in bridge and hockey. The other models could not interpret bridge as a mind sport. BERTje is correlating hockey the highest with football and tennis, probably due to the popularity of the sports. The last sport, ‘bergsport’ was only correlating with words that also had sport in it. This can be caused by not being in the vocabulary and falling back on word parts.

The last category are the instruments, that are judged on the following instruments; zither, trumpet, recorder and synthesizer. For all the instruments, the models were performing poorly. Even a trumpet, a widely known instrument, does not work well and is not getting close to other wind instruments. Only FastText could disambiguate some instruments and create some good correlations.

4.4 Conclusion

To conclude, the overall qualitative results were not as expected. The quantitative tests were hopeful for well-performing models on the Dutch language in the domain of interests of adolescents. The different tests have shown that the embeddings for interests are not good enough to make meaningful relations. The reasoning behind this conclusion will be expanded on in the Discussion, chapter 5.

Chapter 5

Discussion

This section will discuss the outcomes of the research and offer a reflection, limitations and future research. Firstly, section 5.1 gives a summary of the research, stating the problem statement, the research approach, method, execution and lastly the results. This section will answer the research questions respectively. The summary is followed by a section that reflects on the study. This section elaborates on the course of events that has led to this thesis. The third section, section 5.4, will go into the limitations of the study. Section 5.5 will expand on the future research that. Finally, the conclusions are given in section 5.2.

5.1 Summary

This research focuses on the problem of analysing amounts of unstructured text, in the form of short, colloquial Dutch written by adolescents. For solving this problem, NLP models are used that have proven their worth on the English language, but have counterparts pre-trained on Dutch data. Based on a literature study, the models were selected in combination with the fundamental different neural network architectures and input types the models have (character-, word- and sentence based). Additionally, enrichment strategies are identified, where LM fine-tuning is applied. The data for fine-tuning the models is scraped from social media website Reddit and applied to all the models that are capable of LM fine-tuning. This results in seven models that are tested.

The models are tested in various ways, consisting of two fine-tuning tasks (CoNLL 2002 NER and DBRD SA) for a quantitative comparison and three qualitative tests focusing on the output of the embedding layer of the models. To expand on the qualitative tests, the first test's goal is to see if a model can correctly rank the correlation scores between a triplet of real-data interests. The second test is a inductive analysis where experts evaluate the models capabilities through a visualization and clustering of interest data. The last test is a comparison between hierarchically structured data from Wikipedia that the model has to replicate. The outcome of the qualitative tests helps to define if the models create meaningful representations of interest data and how those representations relate to each other.

This model test design leads to a thorough evaluation of the models where a combination of quality of the embeddings and the results of downstream tasks conclude into an answer to the following question. What NLP techniques can be applied and perform well in order to analyse Dutch interest data captured over time while accounting for colloquial language used by adolescents?

Unfortunately, the models do not perform up to the standards that were expected. The models show promising scores on the quantitative analyses. However, the results of the qualitative tests disappoint. The models have difficulties to correlate related interests together and do not agree most of the time with expert evaluated rankings. The second test shows that models are not able to relate similar interests together on their meaning, but find relatedness strongly in matching word parts. This observation is confirmed in the third test, where correlation scores are high for matching word parts, but for similar instruments or sports, it is not.

For using the models, a tool is created that consists out of two parts. The first part is the 'embedder', where the data is embedded and ready for use in the second part. The second part can perform multiple automated functions such as reduce and visualize the data as a plot, cluster your data, or give the correlation scores of the data. A separate instance is made for modelling the data of interests over time.

5.2 Conclusions

By going through the research questions chronologically, and answering them respectively, this research can come to its final conclusions. The first sub-RQ "How do different (pre-trained) models relate to each other in terms of performance?", has been answered in stages. Firstly, the literature study identified NLP models that could be identified as candidates. The native Dutch models were found to be outperforming multilingual models. Secondly, the selected models were evaluated on a broad set of tasks in order to quantitatively and qualitatively compare the models. The relation of the nature of the models to the nature of the data also plays a part. To answer the question, BERTje performs best on the quantitative tasks and the performance on the qualitative tests is rated poorly for all models.

The second sub-RQ is stated as follows: "How to account for the pipeline in the use of the Dutch language in modelling personal interests?" This question was also firstly tackled by using the literature study, identifying the possible solutions for the Dutch language. Models have the ability to fine-tune, and even for languages, such as a multilingual pre-trained version of BERT (Devlin et al., 2018) shows. Furthermore, there are native Dutch, so pre-trained model trained on only Dutch text, available. As identified earlier, the interest data contains misspelling, slang and colloquial Dutch, written by adolescents. To account for this, models LM fine-tuned on Reddit data are used in the model comparison, unfortunately not showing big differences with the models that are not fine-tuned.

The last RQ is stated as the following: "What ways can the models be visualized and how to interpret the results?" This question focuses on the implementation of the tool. The tool is the basis for the visualization and interpretation of the results, serving as a gateway. The tool consists of different options for visualization such as plotting through dimension reduction, adding a clustering layer over that and cosine similarity scores. The tool is easy to use, intuitive and mainly kept very simple. The only downside is, due to technology constraints, that the embedding step is separate of the visualizations.

To conclude, answering the main research question "What NLP techniques can be applied and perform well in order to analyse Dutch interest data captured over time and how to account for colloquial language used by adolescents? The use of language models that are pre-trained on the Dutch language are available. While the performance on downstream tasks such as NER and sentiment analysis are good, the outputs of the embedding layer are disappointing. The language models seem to have no understanding of the semantical aspect of the Dutch language and do not perform on semantical relation classification and clustering of interests. Fine-tuning seems to help the models to get a grasp of the interest data, but is model specific.

5.3 Reflections

Over the last year, while the study was conducted, a lot of revelations have occurred during the exploratory- , conducting- and reviewing stages. During the exploratory phase, the focus lied on the SOTA of NLP models, their capabilities and shortcomings. While reading the different papers, the gist was that the pre-training of the models would result in a neural network that provides a semantical representation of the language trained on (Akbik et al., 2018; M. Peters et al., 2018).

With this assumption, the research has been conducted and the focus has been on the difference between the architectural choices from the models within. The difference between character-, word- and sentence based models has led the researchers to believe to interact differently with the target task, analysing interest data of short nature, with special focus on spelling mistakes and a relatively new vocabulary.

While conducting the experiments, The first model that worked was the FastText model, plotting interests in a two-dimensional plot using PCA as dimension reduction technique. The result was promising, showing clusters like school courses and sports. The overall consensus was at that moment was; FastText is the first and oldest model, trained on a relatively small dataset (only Wikipedia) and has no contextual awareness. Therefore, the use of the Flair and Transformer models will likely improve on this. This was the assumption made at that moment in time.

With the design of the qualitative tests running longer than expected and coming up with new questions while designing, the quantitative tests were ran first. Again, with very promising results, showing scores from BERT that were not expected in a positive way. For the other models, the expectation that Flair would perform well on the NER task was validated and showing more difficulties on the sentiment analysis task was expected. This was the same for the FastText model, still putting up decent scores on these tasks.

On the contrary, the triplets scores were not what was expected. It was not an outrageous expectation or standards set too high, but the best model only having less than 30% correct was truly striking. Subsequently, the task was reviewed and the researchers concluded that the task is fairly hard for a model to do correctly. The differences between the interests in the triplets are subtle and from different perspective and reasoning, it is possible to come to other conclusions and thus, other rankings. The ranking has been an intensive process where experts in the field of interest analysis have conducted the analysis of the triplets. The structure on how interests are reviewed is substantial and therefore this test and its results are not nullified. Actually, the test shows the intricate nature of the Dutch language or even language in general. Therefore, this task can be provided as a general benchmark on how the relation structure of interests are expressed in language models.

It was up and until the qualitative tasks before the drawbacks became clear for using language models. While conducting the cluster analysis, the results were not as expected. To be blunt, far from what was expected. The nature of the embeddings was not at all what was expected and showed the superficial understanding of language. The clusters were partially correct, but were more morphologically based than semantically based. There was doubt in the use of the dimension reduction- and clustering technique used, however, the cosine similarity scores were rejecting this doubt immediately.

To confirm our suspicion that the problems were originating from the nature of the embeddings, the third test felt as a painful defeat of the language models. Where the given structures were expected to be known by the models such as common sports and instruments, all the models did not succeed in doing so. Again, the models seemed to focus on the morphological structures and not the semantics of the middle school courses, sports and instruments.

To conclude, the purpose of the reflection has been to contextualize the research approach and the unexpected underestimation of the NLP models. The late realisation of the performance of the model has resulted in a lack of applying data enrichment and knowledge infusion. Section 5.5 describes the different steps that can be taken to improve the results from a data-centered approach.

5.4 Limitations

The limitations of this project are discussed using the four aspects of validity (Wohlin et al., 2012). for the inductive interest analysis and clustering, questions were raised if the dimension reduction and clustering techniques did not alter the interests in such a way that signal was lost. To overcome this, an exploratory analysis of the correlation scores of the interest data was done (section 4.3.2). The results of this analysis found that the correlations scores, based on raw embeddings, were in line with the reduced and clustered data.

The conclusion validity of this research could suffer through the sampling of results of the deductive interest analysis. The samples were consistent over all models, but these samples could have had an influence on the impression of the quality on the researcher. All data could be analysed, however in the constraint of time, this is not a possibility.

For construct validity, this research could look into mono-method bias. The only method that is used to depict the relation between interests are NLP models. However, there are more options that could be used to do this, such as probabilistic techniques. Again this brings their own set of challenges and literature shows that these relations can be mapped quite nicely with NLP models (Mikolov, Sutskever, et al., 2013).

Lastly, general remarks that can be made on the experimental setup of the quantitative tests is the difference in training methods. For training and fine-tuning the Flair and FastText model, the Flair framework (Akbik et al., 2019) is used. For training and fine-tuning of the Transformer based models, so BERTje and RobBERT, Huggingface Transformers is used. While the models use different frameworks, the underlying implementation both rely on the PyTorch framework. While there are differences in usage and method, both framework's underlying foundation is equal, hopefully therefore not differing too much in results, but a factor to keep in mind.

5.5 Future research

This research has shown that there is still a lot to gain on the performance for pre-trained NLP models on the Dutch language. The identified cause for the lack of performance is the quality of embeddings demonstrated by the qualitative tests executed in this research. The different models lack the ability to create relations based on meaning and quickly fall back on similarity of words. Therefore, it would be interesting to further evaluate the embeddings. This research should be done from a more general perspective, not focused on interest analysis, and ideally provide a framework for analysis, comparing and contrasting model embedding quality.

This research uses the vector representations for measuring the relation to other words. Thereafter, clusters are created and plotting overall distances is used to analyse relations. An unexplored option is to create a downstream task for this. For example, the clustering method can be generalised to a multi-classification problem, where the models are presented with a training set in order to learn the predefined classes (clusters) of interests.

Lastly, the models and data could be enriched. From a data-centered approach, the interest fields can be enhanced. There is extra data available that describe the interests in more detail. These fields can be added as a whole to the input but a more delicate approach would be using only certain parts of the sentence that is interesting. This can be done through a NER system that selects certain named entities to add to the input. The model enrichment can also be done through knowledge infusion. This can be done in multiple ways, through fully training a neural network on knowledge graphs, such as ERNIE (Zhang et al., 2019). Secondly, this can be done through taking a pre-trained model and retraining it, for example KnowbERT (M. E. Peters et al., 2019). The last option is training a smaller neural network on knowledge data and fusing this together with a pre-trained model, therefore not having to change the pre-trained model. This method is called K-adapter (R. Wang et al., 2020).

Bibliography

Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., & Vollgraf, R. (2019). FLAIR: An Easy-to-Use Framework for State-of-the-Art NLP. *Proceedings of the 2019 Conference of the North*, 54–59. https://doi.org/10.18653/v1/N19-4010

Akbik, A., Blythe, D., & Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, 1638–1649. http://aclweb.org/anthology/C18-1139

Akkerman, S. F., & Bakker, A. (2019). Persons pursuing multiple objects of interest in multiple contexts. *European Journal of Psychology of Education*, *34*(1), 1–24. https://doi.org/10.1007/s10212-018-0400-2

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, *5*arXiv 1607.04606, 135–146. https://doi.org/10.1162/tacl_a_00051

Collobert, R., & Weston, J. (2008). A unified architecture for natural language processing, 160–167. https://doi.org/10.1145/1390156.1390177

de Vries, W., van Cranenburgh, A., Bisazza, A., Caselli, T., van Noord, G., & Nissim, M. (2019). BERTje: A Dutch BERT Model, arXiv 1912.09582. http://arxiv.org/abs/1912.09582

Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT: a Dutch RoBERTa-based Language Model, arXiv 2001.06286. http://arxiv.org/abs/2001.06286

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2018). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding, (1950), arXiv 1810.04805. http://arxiv.org/abs/1810.04805

Hochreiter, S., & Schmidhuber, J. (1997). LSTM can solve hard long time lag problems. *Advances in Neural Information Processing Systems*, 473–479.

Howard, J., & Ruder, S. (2018). Universal language model fine-tuning for text classification. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*arXiv 1801.06146, 328–339. https://doi.org/10.18653/v1/p18-1031

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., & Wu, Y. (2016). Exploring the Limits of Language Modeling, arXiv 1602.02410. http://arxiv.org/abs/1602.02410

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., & Stoyanov, V. (2019). RoBERTa: A Robustly Optimized BERT Pretraining Approach, (1), arXiv 1907.11692. http://arxiv.org/abs/1907.11692

Matthews, B. W. (1975). (Received May 21st, 1975), *405*, 442–451.

McInnes, L., Healy, J., & Astels, S. (2017). hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, *2*(11). https://doi.org/10.21105/joss.00205

McInnes, L., Healy, J., Saul, N., & Grossberger, L. (2018). UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software*, *3*(29), 861.

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space, arXiv 1301.3781, 1–12. http://arxiv.org/abs/1301.3781

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., & Dean, J. (2013). Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, arXiv 1310.4546, 1–9.

Pennington, J., Socher, R., & Manning, C. D. (2014). GloVe: Global vectors for word representation. *EMNLP 2014 - 2014 Conference on Empirical Methods in Natural Language Processing, Proceedings of the Conference*, 1532–1543. https://doi.org/10.3115/v1/d14-1162

Peters, M. E., Ammar, W., Bhagavatula, C., & Power, R. (2017). Semi-supervised sequence tagging with bidirectional language models. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, *1*arXiv 1705.00108, 1756–1765. https://doi.org/10.18653/v1/P17-1161

Peters, M. E., Neumann, M., Logan, R., Schwartz, R., Joshi, V., Singh, S., & Smith, N. A. (2019). Knowledge Enhanced Contextual Word Representations, arXiv 1909.04164, 43–54. https://doi.org/10.18653/v1/d19-1005

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., & Zettlemoyer, L. (2018). Deep Contextualized Word Representations, arXiv 1802.05365, 2227–2237. https://doi.org/10.18653/v1/n18-1202

Radford, A., & Salimans, T. (2018). Improving Language Understanding by Generative Pre-Training (transformer in real world). *OpenAI*, 1–12. https://s3-us-west-2.amazonaws.com/openai-assets/research-covers/language-unsupervised/language%7B%5C_%7Dunderstanding%7B%5C_%7Dpaper.pdf

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., & Sutskever, I. (2018). Language Models are Unsupervised Multitask Learners.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., & Liu, P. J. (2019). Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer, arXiv 1910.10683, 1–53. http://arxiv.org/abs/1910.10683

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature*, *323*(6088), 533–536. https://doi.org/10.1038/323533a0

Sanh, V., Debut, L., Chaumond, J., & Wolf, T. (2019). DistilBERT, a distilled version of BERT: smaller, faster, cheaper and lighter, arXiv 1910.01108, 2–6. http://arxiv.org/abs/1910.01108

Tjong Kim Sang, E. F. (2002). Memory-Based Named Entity Recognition, In *Proceedings of the 6th conference on natural language learning - volume 20*, USA, Association for Computational Linguistics. https://doi.org/10.3115/1118853.1118878

Van Der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. *Journal of Machine Learning Research*, *9*, 2579–2625.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*, *2017-Decem*(Nips), arXiv 1706.03762, 5999–6009.

Wang, A., Pruksachatkun, Y., Nangia, N., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2019). SuperGLUE: A Stickier Benchmark for General-Purpose Language Understanding Systems, *2019*(July), arXiv 1905.00537, 1–30. http://arxiv.org/abs/1905.00537

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. (2019). GLUE: A Multi-Task Benchmark and Analysis Platform for Natural Language Understanding, arXiv 1804.07461, 353–355. https://doi.org/10.18653/v1/w18-5446

Wang, R., Tang, D., Duan, N., Wei, Z., Huang, X., Ji, J., Cao, G., Jiang, D., & Zhou, M. (2020). K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters, arXiv 2002.01808. http://arxiv.org/abs/2002.01808

Wieringa, R. J. (2014). *Design science methodology: For information systems and software engineering*. https://doi.org/10.1007/978-3-662-43839-8

Wohlin, C., Runeson, P., Hst, M., Ohlsson, M. C., Regnell, B., & Wessln, A. (2012). *Experimentation in Software Engineering*. Springer Publishing Company, Incorporated.

Wold, S., Esbensen, K., & Geladi, P. (1987). Principal component analysis. *Chemometrics and Intelligent Laboratory Systems*, *2*(1), 37–52. https://doi.org/https://doi.org/10.1016/0169-7439(87)80084-9
Proceedings of the Multivariate Statistical Workshop for Geologists and Geochemists

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., & Brew, J. (2019). HuggingFace's Transformers: State-of-the-art Natural Language Processing. *ArXiv*, *abs/1910.0*.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., & Le, Q. V. (2019). XLNet: Generalized Autoregressive Pretraining for Language Understanding, arXiv 1906.08237, 1–18. http://arxiv.org/abs/1906.08237

Zhang, Z., Han, X., Liu, Z., Jiang, X., Sun, M., & Liu, Q. (2019). ERNIE: Enhanced Language Representation with Informative Entities, arXiv 1905.07129, 1441–1451. https://doi.org/10.18653/v1/p19-1139

Appendix A

This appendix elaborates on the research protocol that is the inception for the literature review. The motivation is described in section A.1 accompanied with a flow chart of the process (figure A.1).

A.1 Literature research protocol

Research in the realm of NLP is a challenge in multiple aspects. There is a vast difference in literature research with other fields, and therefore, chosen for an alternative research protocol. Firstly, the field of NLP is moving very rapidly. Since the inception of efficient pre-trained word embeddings for neural networks (Mikolov, Sutskever, et al., 2013), the pace increased to multiple new SOTA language models per year. Not only the push of new scientific work has brought the field so far, the pull of corporate- and consumer applications has created demand for technological advances in this field. Research institutes from the biggest conglomerates around the world (e.g. Facebook, Google, Tencent, Zalando and others) have had their fair share in contributing to research and technological advances in NLP (Akbik et al., 2018; Bojanowski et al., 2017; Devlin et al., 2018). Next to universities and corporates, other research institutes such as AllenNLP (M. Peters et al., 2018) and Openai (Radford et al., 2018) have contributed to this field.

This vast amount of researchers working in the field of NLP, and therefore the sheer amount of papers published, has an effect on the literature research method. Using search engines such as Arxiv or Google Scholar with chosen keywords have adversarial effects. Firstly the number of papers that are returned in keyword search is immense. Making a selection or a cut-off at a certain metric (number of citations etc.) would be arbitrary. Secondly, the quality of the papers found in the search engines is sometimes hard to determine. Papers can be pre-published before peer review or the relevancy is not yet established.

This can be partially controlled by using a different source to start to dive into NLP research by making sure the quality and impact of papers are substantial. In this research protocol, the use of the association of computer linguistics (ACL) conference proceedings will be the beginning of creating a thorough understanding of the field of NLP. Using the last three years (2019, ’18 and ‘17) of the ACL proceedings, 2513 papers in total of which 748 are long/short papers and the rest workshops or demonstrations. These papers are thoroughly reviewed and therefore ensured of high quality. Secondly, the yearly conference ensures that the papers are relevant for their time and not obsolete when published.

By creating a high quality subset of the available research papers, the next step in the literature research protocol is selecting the relevant papers. Using different keyword(s) on the abstracts of the papers, a filter on the papers is applied to find papers with coinciding interests of research. The abstracts with one or more keyword(s) are then read and selected/discarded on relevance according to the researcher. This final selection of papers lays the groundwork for the literature research of this thesis. Figure 2 depicts the flow of the literature research protocol.

To accompany the first method described above, (reverse) snowballing is applied to dive deeper into certain subjects that sparked interest and/or were deemed fruitful for this research. Furthermore, papers that were referenced through multiple papers were marked as fundamental for certain techniques and therefore important for further research.

Finally, the field of NLP is marked by an approach of practicality and papers are more often than not accompanied by links to code repositories with implementations available. This makes that sources that are not necessarily peer reviewed papers are also in need of attention to fully grasp the impact and degree of importance of some papers. Practitioners of this field hold valuable information in the form of blogs and accounts on for example Medium, where they inform a wide ranging audience on NLP papers, techniques and methods that are relevant today. These blogs are not used as sources but rather as inspiration and understanding of certain techniques and are found helpful to progress the search of literature throughout.

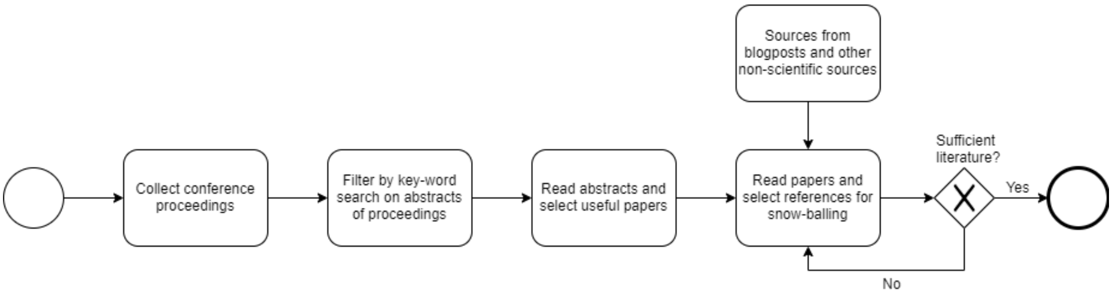
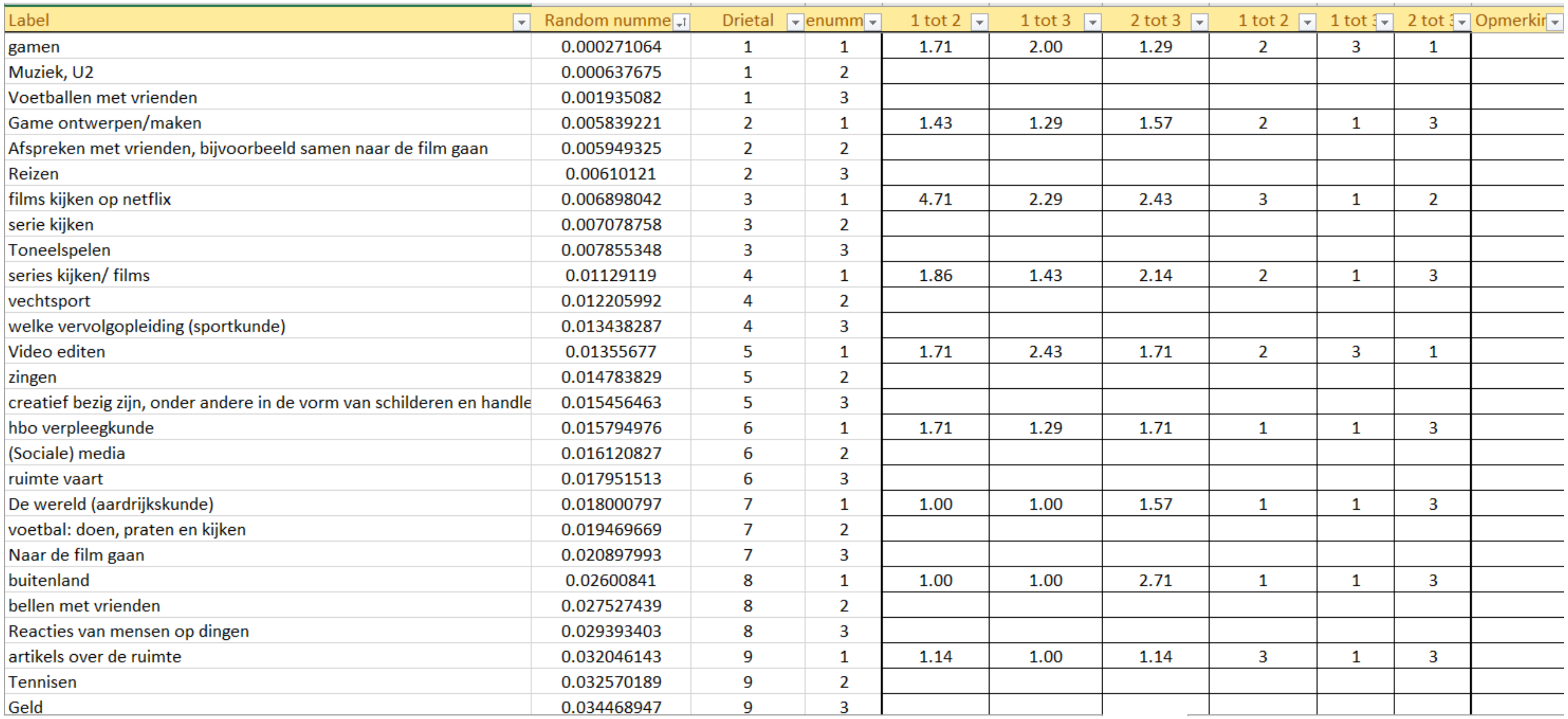

Sources from blogposts and other non-scientific sources
Collect conference proceedings
Filter by key-word search on abstracts of proceedings
Read abstracts and select useful papers
Read papers and select references for snow-balling
Sufficient literature?
Yes
No

Figure A.1: Diagram of the literature research protocol

Appendix B

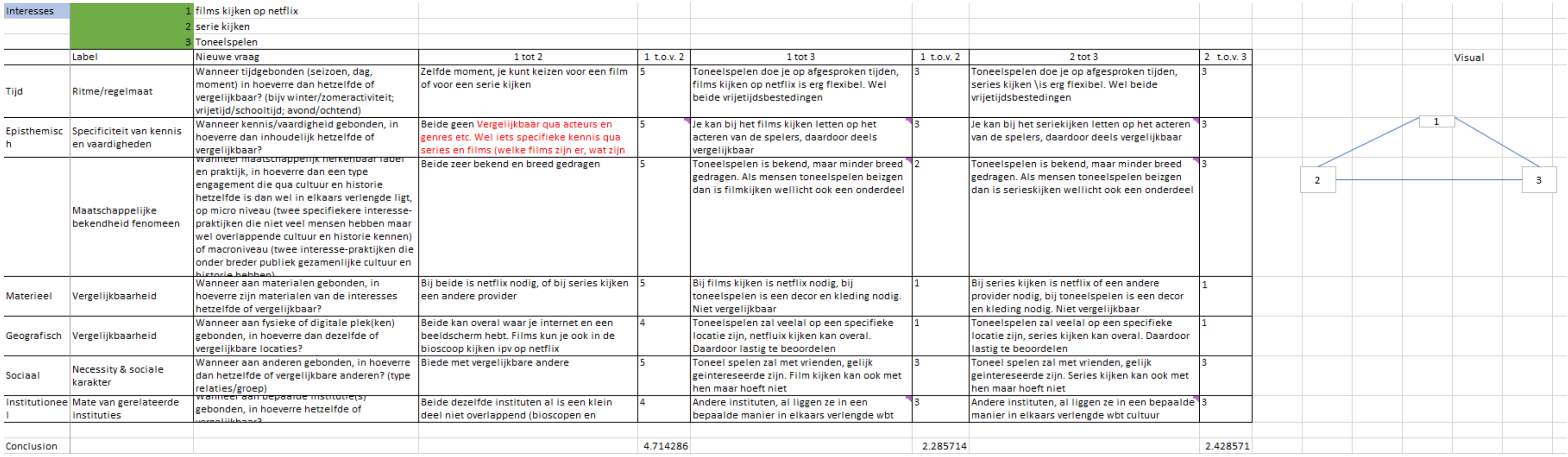
This appendix provides the materials used in the triplets analysis (section 3.4.1). This appendix contains a part of the triplet evaluation done by the experts and a figure containing the results from the models. Note that this is just a part of the spreadsheet containing the analysis. The full spreadsheets are provided through the dedicated repository of this work[1].

[1] https://git.science.uu.nl/tvdermeer/thesis



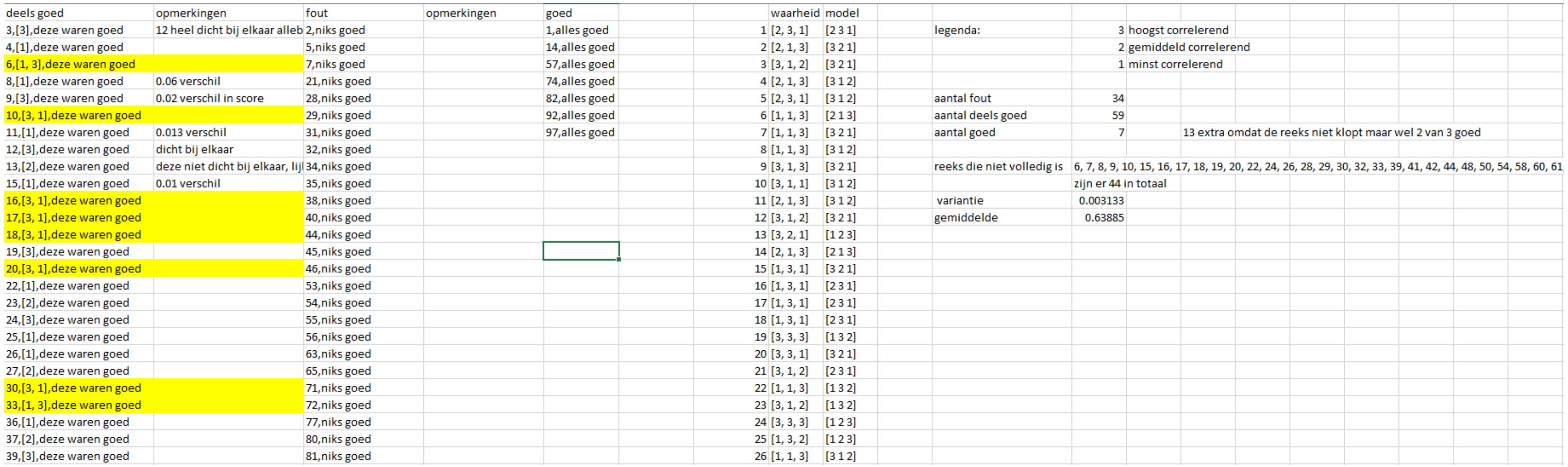
Label	Random numme	Drietal	enumm	1 tot 2	1 tot 3	2 tot 3	1 tot 2	1 tot	2 tot	Opmerkir
gamen	0.000271064	1	1	1.71	2.00	1.29	2	3	1	
Muziek, U2	0.000637675	1	2							
Voetballen met vrienden	0.001935082	1	3							
Game ontwerpen/maken	0.005839221	2	1	1.43	1.29	1.57	2	1	3	
Afspreken met vrienden, bijvoorbeeld samen naar de film gaan	0.005949325	2	2							
Reizen	0.00610121	2	3							
films kijken op netflix	0.006898042	3	1	4.71	2.29	2.43	3	1	2	
serie kijken	0.007078758	3	2							
Toneelspelen	0.007855348	3	3							
series kijken/ films	0.01129119	4	1	1.86	1.43	2.14	2	1	3	
vechtsport	0.012205992	4	2							
welke vervolgopleiding (sportkunde)	0.013438287	4	3							
Video editen	0.01355677	5	1	1.71	2.43	1.71	2	3	1	
zingen	0.014783829	5	2							
creatief bezig zijn, onder andere in de vorm van schilderen en handle	0.015456463	5	3							
hbo verpleegkunde	0.015794976	6	1	1.71	1.29	1.71	1	1	3	
(Sociale) media	0.016120827	6	2							
ruimte vaart	0.017951513	6	3							
De wereld (aardrijkskunde)	0.018000797	7	1	1.00	1.00	1.57	1	1	3	
voetbal: doen, praten en kijken	0.019469669	7	2							
Naar de film gaan	0.020897993	7	3							
buitenland	0.02600841	8	1	1.00	1.00	2.71	1	1	3	
bellen met vrienden	0.027527439	8	2							
Reacties van mensen op dingen	0.029393403	8	3							
artikels over de ruimte	0.032046143	9	1	1.14	1.00	1.14	3	1	3	
Tennisen	0.032570189	9	2							
Geld	0.034468947	9	3							

Figure B.1: Triplets analysis of the experts



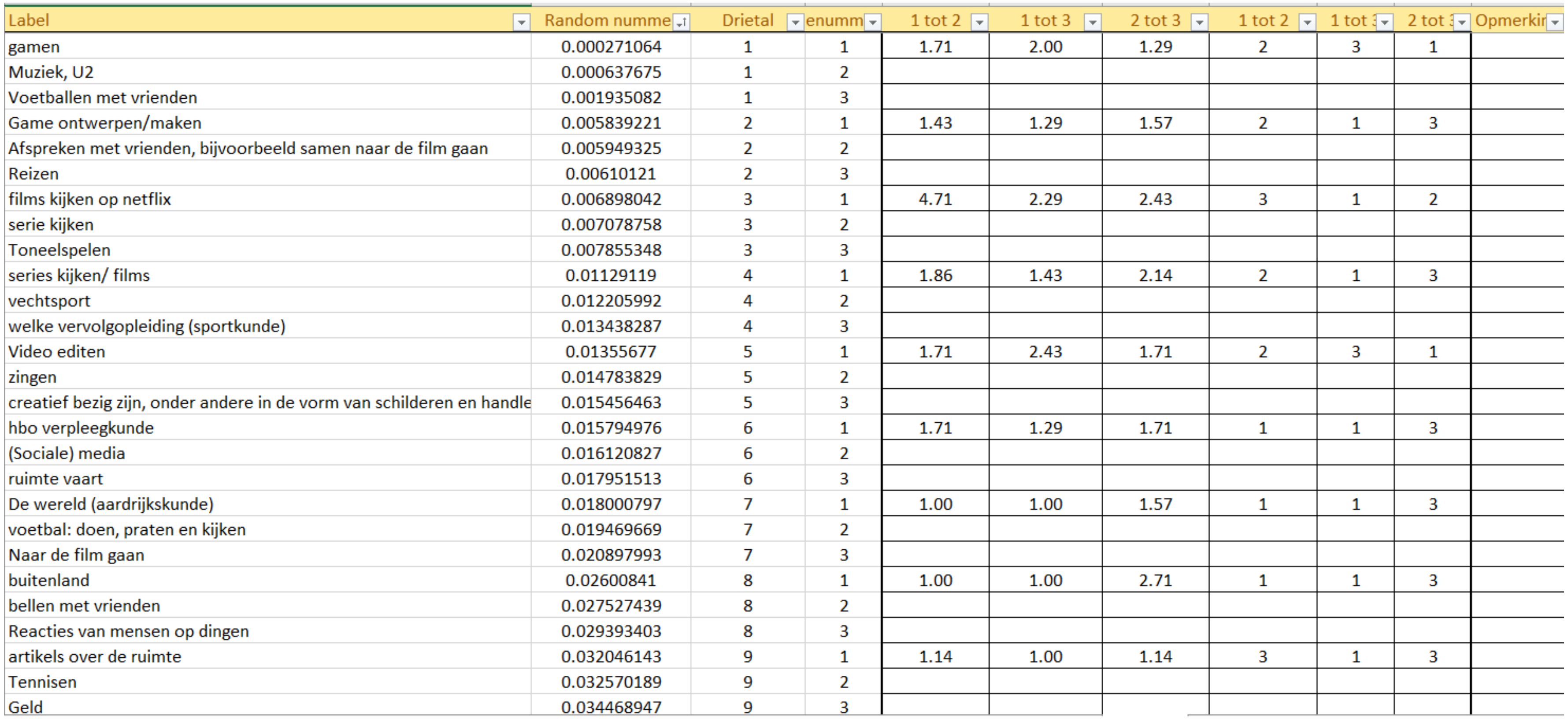
Interesses	1	films kijken op netflix						
	2	serie kijken						
	3	Toneelspelen						
	Label	Nieuwe vraag	1 tot 2	1 t.o.v. 2	1 tot 3	1 t.o.v. 2	2 tot 3	2 t.o.v. 3
Tijd	Ritme/regelmaat	Wanneer tijdgebonden (seizoen, dag, moment) in hoeverre dan hetzelfde of vergelijkbaar? (bijv winter/zomeractiviteit; vrijetijd/schooltijd; avond/ochtend)	Zelfde moment, je kunt keizen voor een film of voor een serie kijken	5	Toneelspelen doe je op afgesproken tijden, films kijken op netflix is erg flexibel. Wel beide vrijetijdsbestedingen	3	Toneelspelen doe je op afgesproken tijden, series kijken \is erg flexibel. Wel beide vrijetijdsbestedingen	3
Episthemisch	Specificiteit van kennis en vaardigheden	Wanneer kennis/vaardigheid gebonden, in hoeverre dan inhoudelijk hetzelfde of vergelijkbaar?	Beide geen Vergelijkbaar qua acteurs en genres etc. Wel iets specifieke kennis qua series en films (welke films zijn er, wat zijn	5	Je kan bij het films kijken letten op het acteren van de spelers, daardoor deels vergelijkbaar	3	Je kan bij het seriekijken letten op het acteren van de spelers, daardoor deels vergelijkbaar	3
	Maatschappelijke bekendheid fenomeen	Wanneer maatschappelijk herkenbaar label en praktijk, in hoeverre dan een type engagement die qua cultuur en historie hetzelfde is dan wel in elkaars verlengde ligt, op micro niveau (twee specifiekere interesse-praktijken die niet veel mensen hebben maar wel overlappende cultuur en historie kennen) of macroniveau (twee interesse-praktijken die onder breder publiek gezamenlijke cultuur en historie hebben)	Beide zeer bekend en breed gedragen	5	Toneelspelen is bekend, maar minder breed gedragen. Als mensen toneelspelen beizgen dan is filmkijken wellicht ook een onderdeel	2	Toneelspelen is bekend, maar minder breed gedragen. Als mensen toneelspelen beizgen dan is serieskijken wellicht ook een onderdeel	3
Materieel	Vergelijkbaarheid	Wanneer aan materialen gebonden, in hoeverre zijn materialen van de interesses hetzelfde of vergelijkbaar?	Bij beide is netflix nodig, of bij series kijken een andere provider	5	Bij films kijken is netflix nodig, bij toneelspelen is een decor en kleding nodig. Niet vergelijkbaar	1	Bij series kijken is netflix of een andere provider nodig, bij toneelspelen is een decor en kleding nodig. Niet vergelijkbaar	1
Geografisch	Vergelijkbaarheid	Wanneer aan fysieke of digitale plek(ken) gebonden, in hoeverre dan dezelfde of vergelijkbare locaties?	Beide kan overal waar je internet en een beeldscherm hebt. Films kun je ook in de bioscoop kijken ipv op netflix	4	Toneelspelen zal veelal op een specifieke locatie zijn, netfluix kijken kan overal. Daardoor lastig te beoordelen	1	Toneelspelen zal veelal op een specifieke locatie zijn, series kijken kan overal. Daardoor lastig te beoordelen	1
Sociaal	Necessity & sociale karakter	Wanneer aan anderen gebonden, in hoeverre dan hetzelfde of vergelijkbare anderen? (type relaties/groep)	Biede met vergelijkbare andere	5	Toneel spelen zal met vrienden, gelijk geintereseerde zijn. Film kijken kan ook met hen maar hoeft niet	3	Toneel spelen zal met vrienden, gelijk geintereseerde zijn. Series kijken kan ook met hen maar hoeft niet	3
Institutioneel	Mate van gerelateerde instituties	Wanneer aan bepaalde institutie(s) gebonden, in hoeverre hetzelfde of vergelijkbaar?	Beide dezelfde instituten al is een klein deel niet overlappend (bioscopen en	4	Andere instituten, al liggen ze in een bepaalde manier in elkaars verlengde wbt	3	Andere instituten, al liggen ze in een bepaalde manier in elkaars verlengde wbt cultuur	3
Conclusion				4.714286		2.285714		2.428571

Figure B.2: Triplets analysis of the experts of a single triplets



deels goed	opmerkingen	fout	opmerkingen	goed			waarheid	model				
3,[3],deze waren goed	12 heel dicht bij elkaar alleb	2,niks goed		1,alles goed		1	[2, 3, 1]	[2 3 1]	legenda:	3	hoogst correlerend	
4,[1],deze waren goed		5,niks goed		14,alles goed		2	[2, 1, 3]	[3 2 1]		2	gemiddeld correlerend	
6,[1, 3],deze waren goed		7,niks goed		57,alles goed		3	[3, 1, 2]	[3 2 1]		1	minst correlerend	
8,[1],deze waren goed	0.06 verschil	21,niks goed		74,alles goed		4	[2, 1, 3]	[3 1 2]				
9,[3],deze waren goed	0.02 verschil in score	28,niks goed		82,alles goed		5	[2, 3, 1]	[3 1 2]	aantal fout	34		
10,[3, 1],deze waren goed		29,niks goed		92,alles goed		6	[1, 1, 3]	[2 1 3]	aantal deels goed	59		
11,[1],deze waren goed	0.013 verschil	31,niks goed		97,alles goed		7	[1, 1, 3]	[3 2 1]	aantal goed	7		13 extra omdat de reeks niet klopt maar wel 2 van 3 goed
12,[3],deze waren goed	dicht bij elkaar	32,niks goed				8	[1, 1, 3]	[3 1 2]				
13,[2],deze waren goed	deze niet dicht bij elkaar, lijl	34,niks goed				9	[3, 1, 3]	[3 2 1]	reeks die niet volledig is	6, 7, 8, 9, 10, 15, 16, 17, 18, 19, 20, 22, 24, 26, 28, 29, 30, 32, 33, 39, 41, 42, 44, 48, 50, 54, 58, 60, 61		
15,[1],deze waren goed	0.01 verschil	35,niks goed				10	[3, 1, 1]	[3 1 2]		zijn er 44 in totaal		
16,[3, 1],deze waren goed		38,niks goed				11	[2, 1, 3]	[3 1 2]	variantie	0.003133		
17,[3, 1],deze waren goed		40,niks goed				12	[3, 1, 2]	[3 2 1]	gemiddelde	0.63885		
18,[3, 1],deze waren goed		44,niks goed				13	[3, 2, 1]	[1 2 3]				
19,[3],deze waren goed		45,niks goed				14	[2, 1, 3]	[2 1 3]				
20,[3, 1],deze waren goed		46,niks goed				15	[1, 3, 1]	[3 2 1]				
22,[1],deze waren goed		53,niks goed				16	[1, 3, 1]	[2 3 1]				
23,[2],deze waren goed		54,niks goed				17	[1, 3, 1]	[2 3 1]				
24,[3],deze waren goed		55,niks goed				18	[1, 3, 1]	[2 3 1]				
25,[1],deze waren goed		56,niks goed				19	[3, 3, 3]	[1 3 2]				
26,[1],deze waren goed		63,niks goed				20	[3, 3, 1]	[3 2 1]				
27,[2],deze waren goed		65,niks goed				21	[3, 1, 2]	[2 3 1]				
30,[3, 1],deze waren goed		71,niks goed				22	[1, 1, 3]	[1 3 2]				
33,[1, 3],deze waren goed		72,niks goed				23	[3, 1, 2]	[1 3 2]				
36,[1],deze waren goed		77,niks goed				24	[3, 3, 3]	[1 2 3]				
37,[2],deze waren goed		80,niks goed				25	[1, 3, 2]	[1 2 3]				
39,[3],deze waren goed		81,niks goed				26	[1, 1, 3]	[3 1 2]				

Figure B.3: Overview of ranking of triplets by BERTje



Label	Random numme	Drietal	enumm	1 tot 2	1 tot 3	2 tot 3	1 tot 2	1 tot 3	2 tot 3	Opmerkir
gamen	0.000271064	1	1	1.71	2.00	1.29	2	3	1	
Muziek, U2	0.000637675	1	2							
Voetballen met vrienden	0.001935082	1	3							
Game ontwerpen/maken	0.005839221	2	1	1.43	1.29	1.57	2	1	3	
Afspreken met vrienden, bijvoorbeeld samen naar de film gaan	0.005949325	2	2							
Reizen	0.00610121	2	3							
films kijken op netflix	0.006898042	3	1	4.71	2.29	2.43	3	1	2	
serie kijken	0.007078758	3	2							
Toneelspelen	0.007855348	3	3							
series kijken/ films	0.01129119	4	1	1.86	1.43	2.14	2	1	3	
vechtsport	0.012205992	4	2							
welke vervolgopleiding (sportkunde)	0.013438287	4	3							
Video editen	0.01355677	5	1	1.71	2.43	1.71	2	3	1	
zingen	0.014783829	5	2							
creatief bezig zijn, onder andere in de vorm van schilderen en handle	0.015456463	5	3							
hbo verpleegkunde	0.015794976	6	1	1.71	1.29	1.71	1	1	3	
(Sociale) media	0.016120827	6	2							
ruimte vaart	0.017951513	6	3							
De wereld (aardrijkskunde)	0.018000797	7	1	1.00	1.00	1.57	1	1	3	
voetbal: doen, praten en kijken	0.019469669	7	2							
Naar de film gaan	0.020897993	7	3							
buitenland	0.02600841	8	1	1.00	1.00	2.71	1	1	3	
bellen met vrienden	0.027527439	8	2							
Reacties van mensen op dingen	0.029393403	8	3							
artikels over de ruimte	0.032046143	9	1	1.14	1.00	1.14	3	1	3	
Tennisen	0.032570189	9	2							
Geld	0.034468947	9	3							

Figure B.4: triplets analysis of the experts

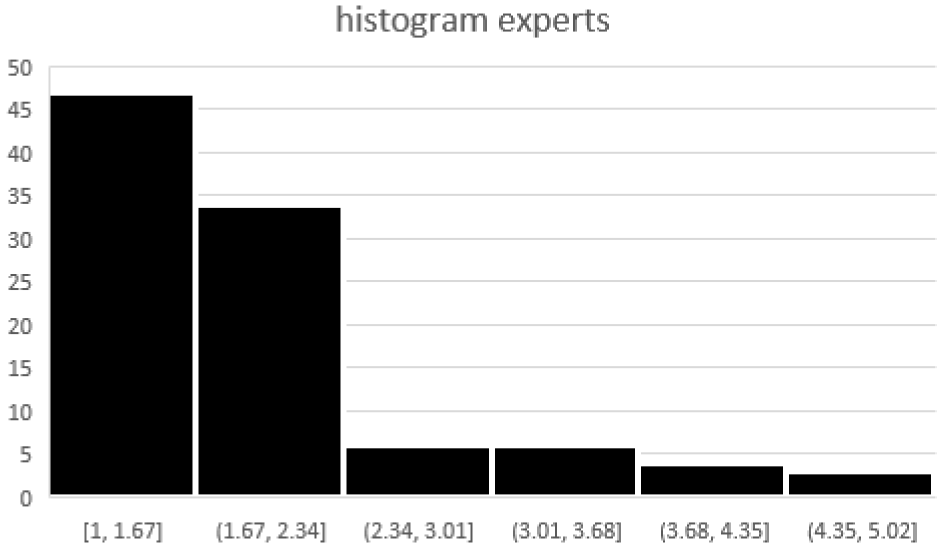


Figure B.5: Histogram of the expert scores distribution

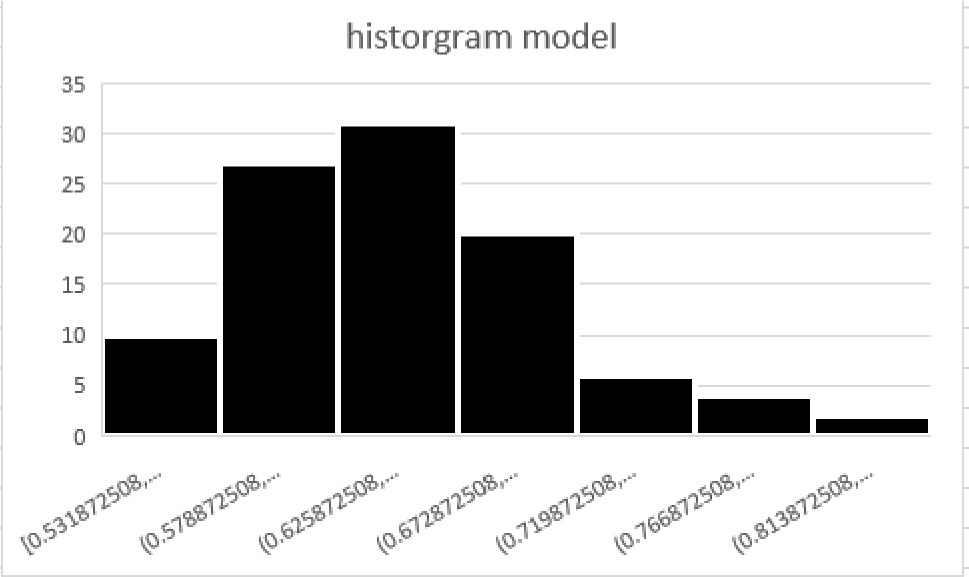


Figure B.6: Histogram of the BERTje model scores distribution

Appendix C

As in appendix B, this appendix contains supporting materials for executing the cluster analysis experiment. Appendix C provides an example of the cluster plot and the spreadsheet used for evaluating one of the models. The additional plots and spreadsheets are found in the repository, noted in appendix B.

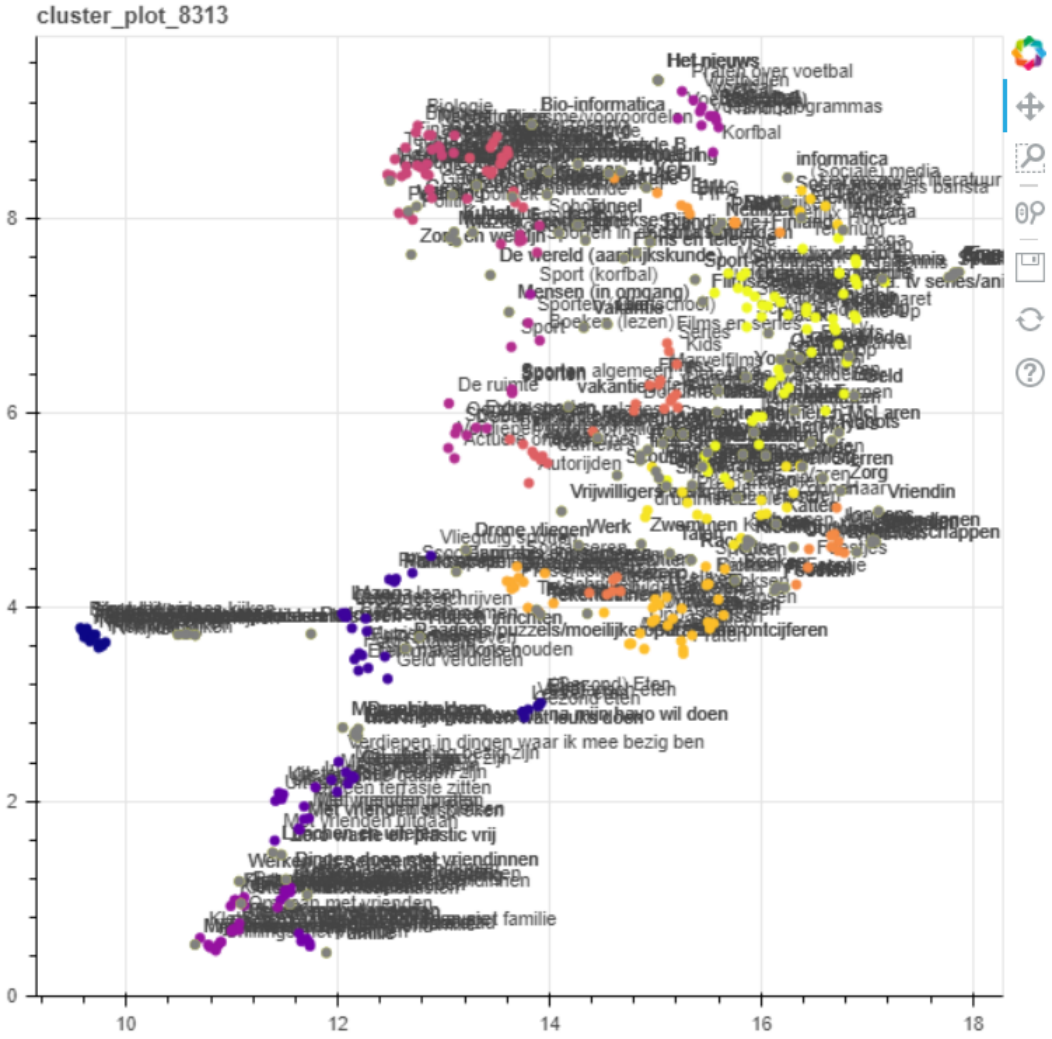
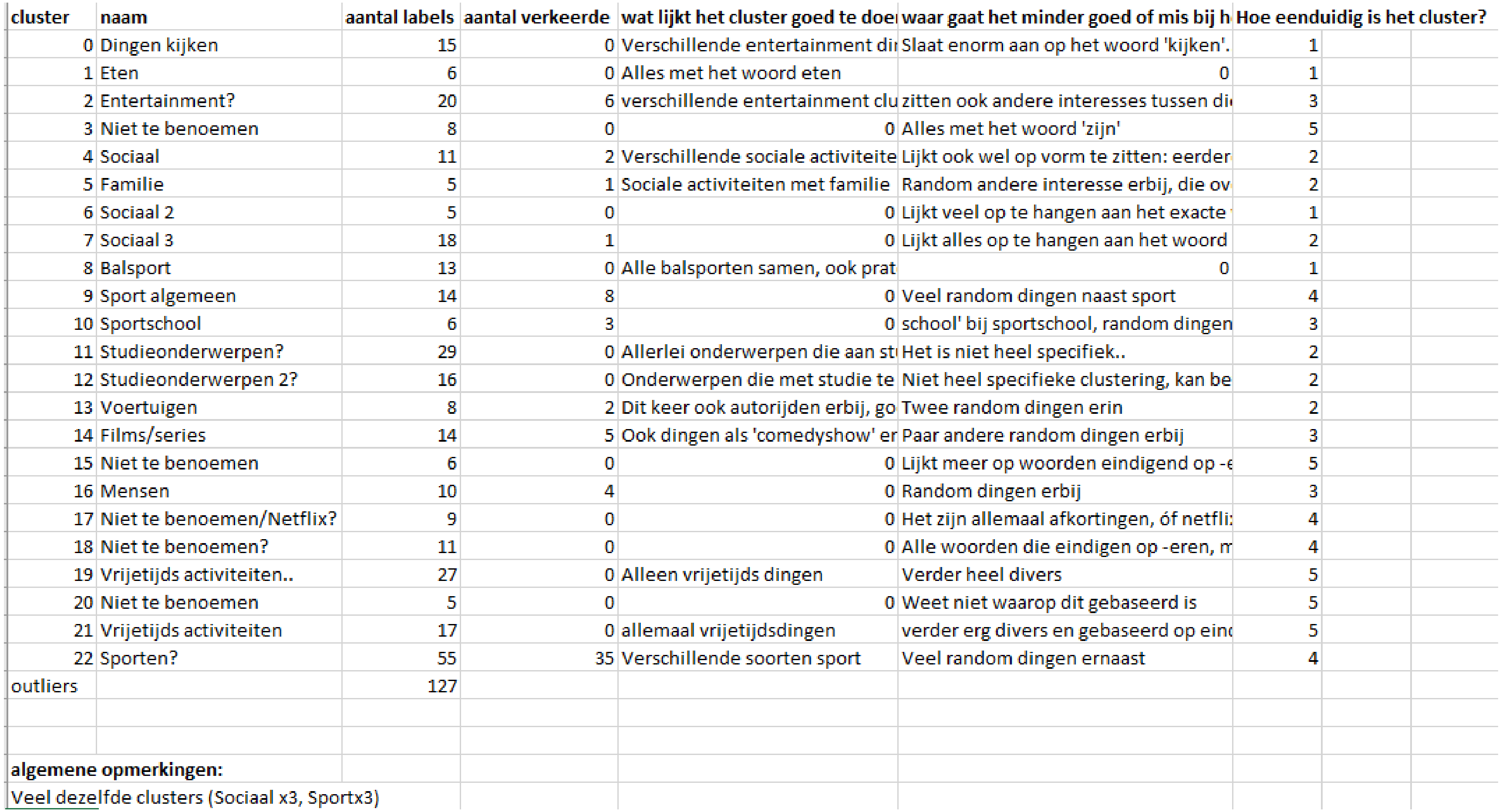
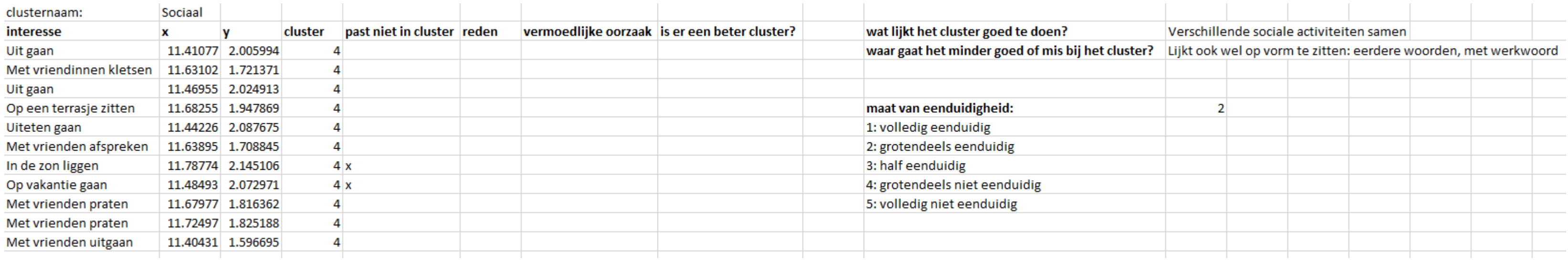


Figure C.1: An interactive cluster plot of the Flair model to investigate the coherence of clusters



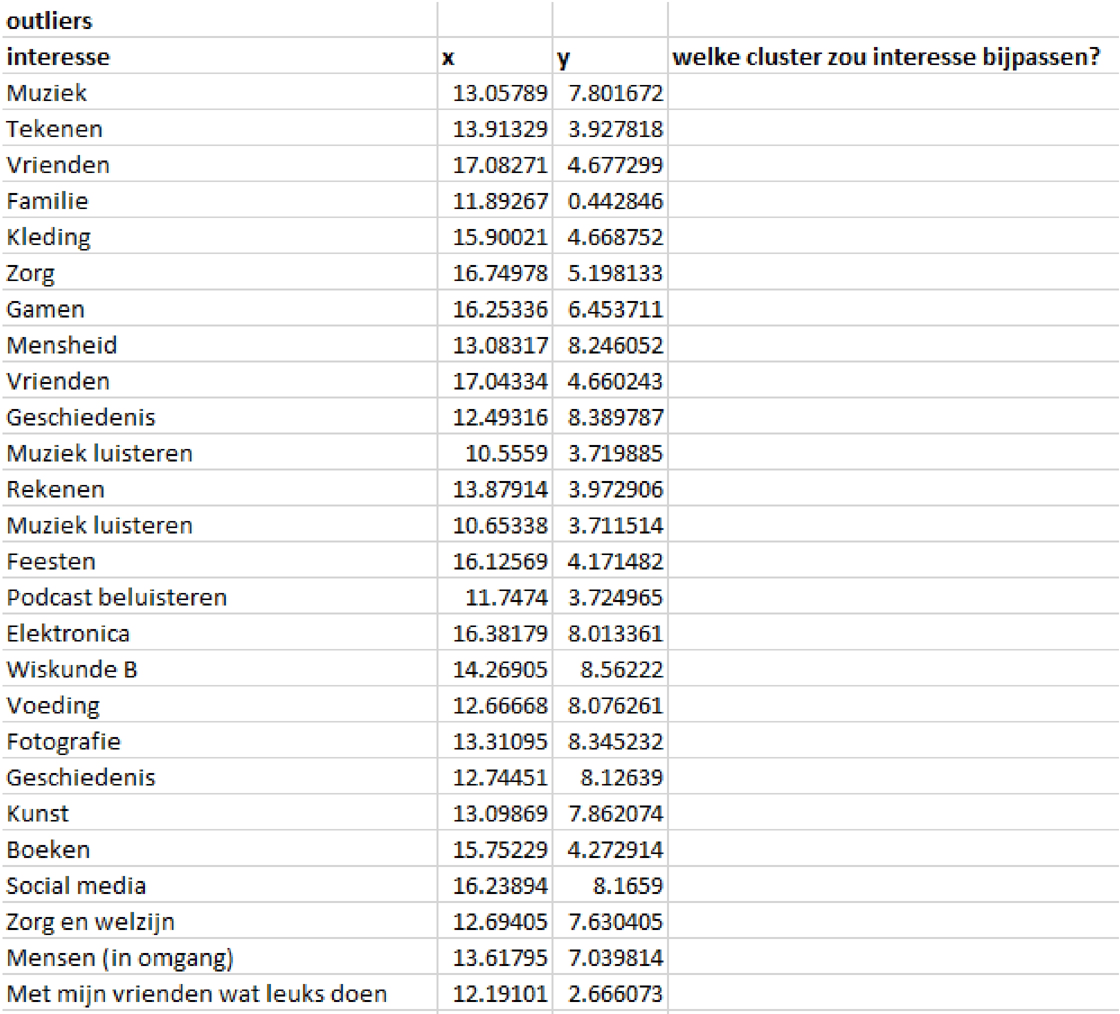
cluster	naam	aantal labels	aantal verkeerde	wat lijkt het cluster goed te doe	waar gaat het minder goed of mis bij h	Hoe eenduidig is het cluster?
0	Dingen kijken	15	0	Verschillende entertainment di	Slaat enorm aan op het woord 'kijken'.	1
1	Eten	6	0	Alles met het woord eten	0	1
2	Entertainment?	20	6	verschillende entertainment clu	zitten ook andere interesses tussen di	3
3	Niet te benoemen	8	0	0	Alles met het woord 'zijn'	5
4	Sociaal	11	2	Verschillende sociale activiteite	Lijkt ook wel op vorm te zitten: eerder	2
5	Familie	5	1	Sociale activiteiten met familie	Random andere interesse erbij, die ov	2
6	Sociaal 2	5	0	0	Lijkt veel op te hangen aan het exacte	1
7	Sociaal 3	18	1	0	Lijkt alles op te hangen aan het woord	2
8	Balsport	13	0	Alle balsporten samen, ook prat	0	1
9	Sport algemeen	14	8	0	Veel random dingen naast sport	4
10	Sportschool	6	3	0	school' bij sportschool, random dingen	3
11	Studieonderwerpen?	29	0	Allerlei onderwerpen die aan st	Het is niet heel specifiek..	2
12	Studieonderwerpen 2?	16	0	Onderwerpen die met studie te	Niet heel specifieke clustering, kan be	2
13	Voertuigen	8	2	Dit keer ook autorijden erbij, go	Twee random dingen erin	2
14	Films/series	14	5	Ook dingen als 'comedyshow' er	Paar andere random dingen erbij	3
15	Niet te benoemen	6	0	0	Lijkt meer op woorden eindigend op -e	5
16	Mensen	10	4	0	Random dingen erbij	3
17	Niet te benoemen/Netflix?	9	0	0	Het zijn allemaal afkortingen, óf netfli	4
18	Niet te benoemen?	11	0	0	Alle woorden die eindigen op -eren, m	4
19	Vrijetijds activiteiten..	27	0	Alleen vrijetijds dingen	Verder heel divers	5
20	Niet te benoemen	5	0	0	Weet niet waarop dit gebaseerd is	5
21	Vrijetijds activiteiten	17	0	allemaal vrijetijdsdingen	verder erg divers en gebaseerd op ein	5
22	Sporten?	55	35	Verschillende soorten sport	Veel random dingen ernaast	4
outliers		127				
algemene opmerkingen:						
Veel dezelfde clusters (Sociaal x3, Sportx3)						

Figure C.2: the cluster overview page of the spreadsheet from the Flair model



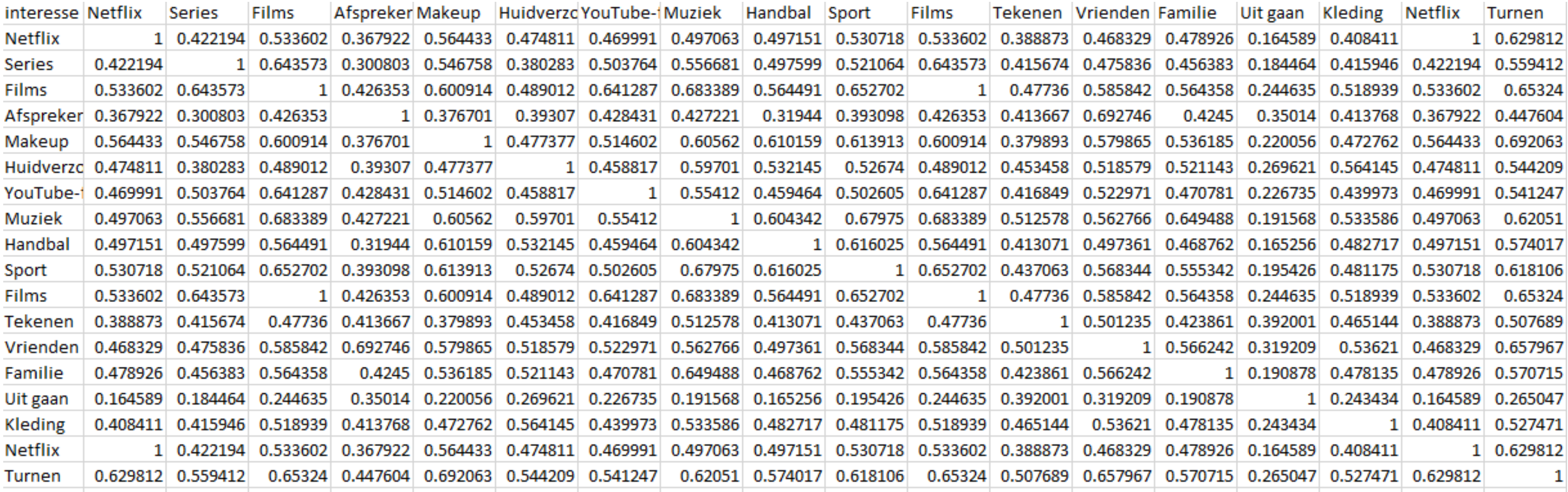
clusternaam:	Sociaal									
interesse	**x**	**y**	**cluster**	**past niet in cluster**	**reden**	**vermoedlijke oorzaak**	**is er een beter cluster?**		**wat lijkt het cluster goed te doen?**	Verschillende sociale activiteiten samen
Uit gaan	11.41077	2.005994	4						**waar gaat het minder goed of mis bij het cluster?**	Lijkt ook wel op vorm te zitten: eerdere woorden, met werkwoord
Met vriendinnen kletsen	11.63102	1.721371	4							
Uit gaan	11.46955	2.024913	4							
Op een terrasje zitten	11.68255	1.947869	4						**maat van eenduidigheid:**	2
Uiteten gaan	11.44226	2.087675	4						1: volledig eenduidig	
Met vrienden afspreken	11.63895	1.708845	4						2: grotendeels eenduidig	
In de zon liggen	11.78774	2.145106	4	x					3: half eenduidig	
Op vakantie gaan	11.48493	2.072971	4	x					4: grotendeels niet eenduidig	
Met vrienden praten	11.67977	1.816362	4						5: volledig niet eenduidig	
Met vrienden praten	11.72497	1.825188	4							
Met vrienden uitgaan	11.40431	1.596695	4							

Figure C.3: The spreadsheet page of one cluster from the Flair model



outliers			
interesse	x	y	welke cluster zou interesse bijpassen?
Muziek	13.05789	7.801672	
Tekenen	13.91329	3.927818	
Vrienden	17.08271	4.677299	
Familie	11.89267	0.442846	
Kleding	15.90021	4.668752	
Zorg	16.74978	5.198133	
Gamen	16.25336	6.453711	
Mensheid	13.08317	8.246052	
Vrienden	17.04334	4.660243	
Geschiedenis	12.49316	8.389787	
Muziek luisteren	10.5559	3.719885	
Rekenen	13.87914	3.972906	
Muziek luisteren	10.65338	3.711514	
Feesten	16.12569	4.171482	
Podcast beluisteren	11.7474	3.724965	
Elektronica	16.38179	8.013361	
Wiskunde B	14.26905	8.56222	
Voeding	12.66668	8.076261	
Fotografie	13.31095	8.345232	
Geschiedenis	12.74451	8.12639	
Kunst	13.09869	7.862074	
Boeken	15.75229	4.272914	
Social media	16.23894	8.1659	
Zorg en welzijn	12.69405	7.630405	
Mensen (in omgang)	13.61795	7.039814	
Met mijn vrienden wat leuks doen	12.19101	2.666073	

Figure C.4: The outlier page of the cluster analysis spreadsheet from the Flair model

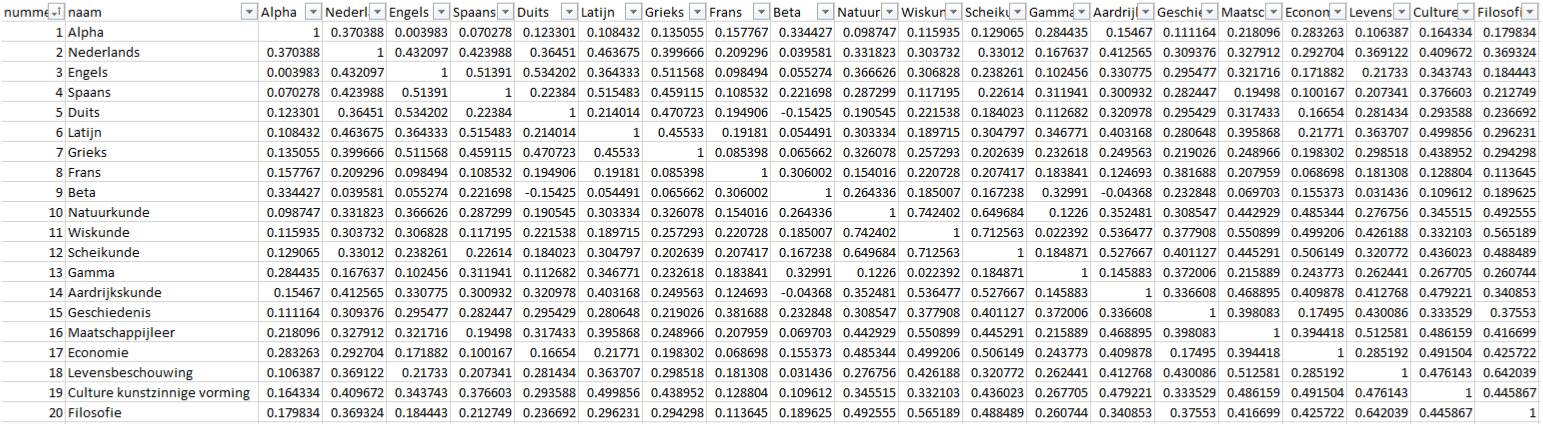


interesse	Netflix	Series	Films	Afspreker	Makeup	Huidverzc	YouTube-t	Muziek	Handbal	Sport	Films	Tekenen	Vrienden	Familie	Uit gaan	Kleding	Netflix	Turnen
Netflix	1	0.422194	0.533602	0.367922	0.564433	0.474811	0.469991	0.497063	0.497151	0.530718	0.533602	0.388873	0.468329	0.478926	0.164589	0.408411	1	0.629812
Series	0.422194	1	0.643573	0.300803	0.546758	0.380283	0.503764	0.556681	0.497599	0.521064	0.643573	0.415674	0.475836	0.456383	0.184464	0.415946	0.422194	0.559412
Films	0.533602	0.643573	1	0.426353	0.600914	0.489012	0.641287	0.683389	0.564491	0.652702	1	0.47736	0.585842	0.564358	0.244635	0.518939	0.533602	0.65324
Afspreker	0.367922	0.300803	0.426353	1	0.376701	0.39307	0.428431	0.427221	0.31944	0.393098	0.426353	0.413667	0.692746	0.4245	0.35014	0.413768	0.367922	0.447604
Makeup	0.564433	0.546758	0.600914	0.376701	1	0.477377	0.514602	0.60562	0.610159	0.613913	0.600914	0.379893	0.579865	0.536185	0.220056	0.472762	0.564433	0.692063
Huidverzc	0.474811	0.380283	0.489012	0.39307	0.477377	1	0.458817	0.59701	0.532145	0.52674	0.489012	0.453458	0.518579	0.521143	0.269621	0.564145	0.474811	0.544209
YouTube-t	0.469991	0.503764	0.641287	0.428431	0.514602	0.458817	1	0.55412	0.459464	0.502605	0.641287	0.416849	0.522971	0.470781	0.226735	0.439973	0.469991	0.541247
Muziek	0.497063	0.556681	0.683389	0.427221	0.60562	0.59701	0.55412	1	0.604342	0.67975	0.683389	0.512578	0.562766	0.649488	0.191568	0.533586	0.497063	0.62051
Handbal	0.497151	0.497599	0.564491	0.31944	0.610159	0.532145	0.459464	0.604342	1	0.616025	0.564491	0.413071	0.497361	0.468762	0.165256	0.482717	0.497151	0.574017
Sport	0.530718	0.521064	0.652702	0.393098	0.613913	0.52674	0.502605	0.67975	0.616025	1	0.652702	0.437063	0.568344	0.555342	0.195426	0.481175	0.530718	0.618106
Films	0.533602	0.643573	1	0.426353	0.600914	0.489012	0.641287	0.683389	0.564491	0.652702	1	0.47736	0.585842	0.564358	0.244635	0.518939	0.533602	0.65324
Tekenen	0.388873	0.415674	0.47736	0.413667	0.379893	0.453458	0.416849	0.512578	0.413071	0.437063	0.47736	1	0.501235	0.423861	0.392001	0.465144	0.388873	0.507689
Vrienden	0.468329	0.475836	0.585842	0.692746	0.579865	0.518579	0.522971	0.562766	0.497361	0.568344	0.585842	0.501235	1	0.566242	0.319209	0.53621	0.468329	0.657967
Familie	0.478926	0.456383	0.564358	0.4245	0.536185	0.521143	0.470781	0.649488	0.468762	0.555342	0.564358	0.423861	0.566242	1	0.190878	0.478135	0.478926	0.570715
Uit gaan	0.164589	0.184464	0.244635	0.35014	0.220056	0.269621	0.226735	0.191568	0.165256	0.195426	0.244635	0.392001	0.319209	0.190878	1	0.243434	0.164589	0.265047
Kleding	0.408411	0.415946	0.518939	0.413768	0.472762	0.564145	0.439973	0.533586	0.482717	0.481175	0.518939	0.465144	0.53621	0.478135	0.243434	1	0.408411	0.527471
Netflix	1	0.422194	0.533602	0.367922	0.564433	0.474811	0.469991	0.497063	0.497151	0.530718	0.533602	0.388873	0.468329	0.478926	0.164589	0.408411	1	0.629812
Turnen	0.629812	0.559412	0.65324	0.447604	0.692063	0.544209	0.541247	0.62051	0.574017	0.618106	0.65324	0.507689	0.657967	0.570715	0.265047	0.527471	0.629812	1

Figure C.5: The similarity matrix for comparing the clusters to the similarity scores for the Flair model

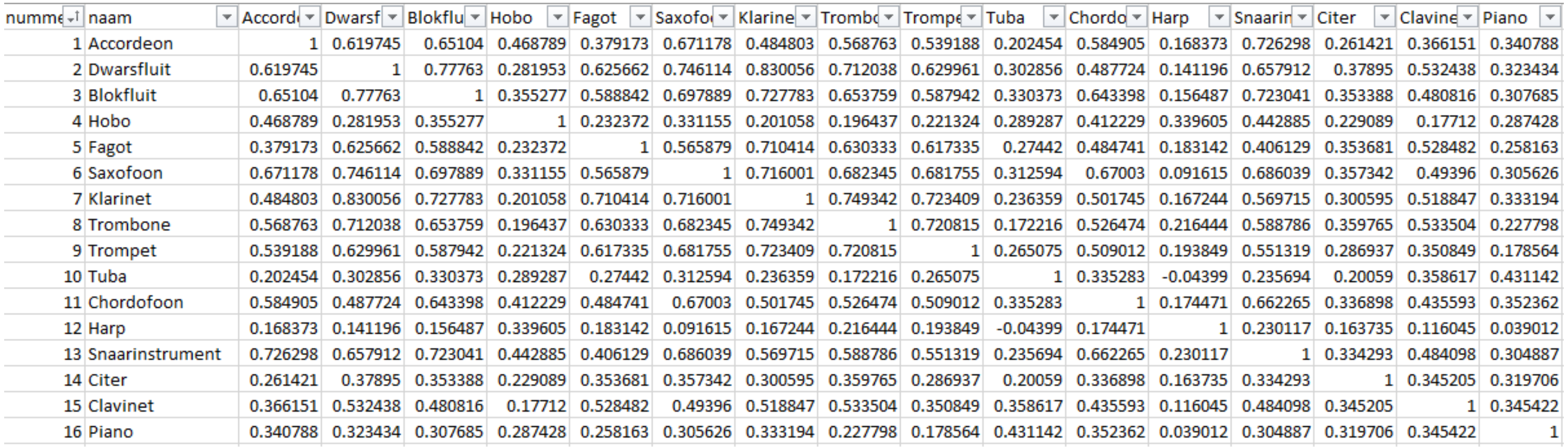
Appendix D

Appendix F contains the spreadsheets used for the deductive analysis. Again, this appendix provides one (part) of the spreadsheet, the full spreadsheets for all the models is found in the repository provided. As a sanity-check and to see if the model BERT multilingual model did not exceed the capabilities of the Dutch models, the spreadsheet for this test is also available in the repository.



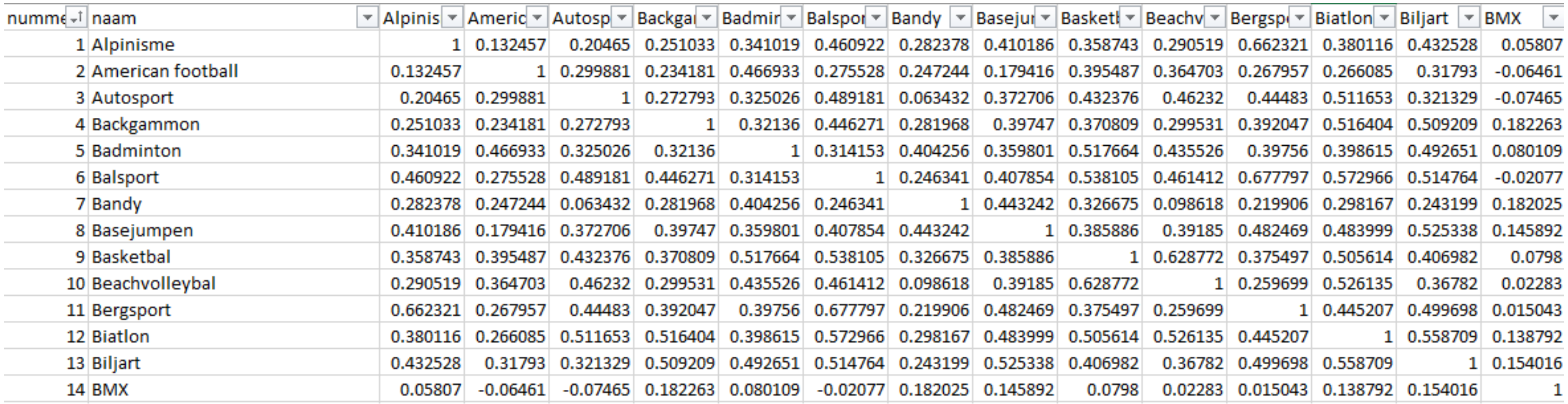
numme	naam	Alpha	Nederl	Engels	Spaans	Duits	Latijn	Grieks	Frans	Beta	Natuur	Wiskur	Scheiku	Gamma	Aardrij	Geschie	Maatsc	Econon	Levens	Culture	Filosofi
1	Alpha	1	0.370388	0.003983	0.070278	0.123301	0.108432	0.135055	0.157767	0.334427	0.098747	0.115935	0.129065	0.284435	0.15467	0.111164	0.218096	0.283263	0.106387	0.164334	0.179834
2	Nederlands	0.370388	1	0.432097	0.423988	0.36451	0.463675	0.399666	0.209296	0.039581	0.331823	0.303732	0.33012	0.167637	0.412565	0.309376	0.327912	0.292704	0.369122	0.409672	0.369324
3	Engels	0.003983	0.432097	1	0.51391	0.534202	0.364333	0.511568	0.098494	0.055274	0.366626	0.306828	0.238261	0.102456	0.330775	0.295477	0.321716	0.171882	0.21733	0.343743	0.184443
4	Spaans	0.070278	0.423988	0.51391	1	0.22384	0.515483	0.459115	0.108532	0.221698	0.287299	0.117195	0.22614	0.311941	0.300932	0.282447	0.19498	0.100167	0.207341	0.376603	0.212749
5	Duits	0.123301	0.36451	0.534202	0.22384	1	0.214014	0.470723	0.194906	-0.15425	0.190545	0.221538	0.184023	0.112682	0.320978	0.295429	0.317433	0.16654	0.281434	0.293588	0.236692
6	Latijn	0.108432	0.463675	0.364333	0.515483	0.214014	1	0.45533	0.19181	0.054491	0.303334	0.189715	0.304797	0.346771	0.403168	0.280648	0.395868	0.21771	0.363707	0.499856	0.296231
7	Grieks	0.135055	0.399666	0.511568	0.459115	0.470723	0.45533	1	0.085398	0.065662	0.326078	0.257293	0.202639	0.232618	0.249563	0.219026	0.248966	0.198302	0.298518	0.438952	0.294298
8	Frans	0.157767	0.209296	0.098494	0.108532	0.194906	0.19181	0.085398	1	0.306002	0.154016	0.220728	0.207417	0.183841	0.124693	0.381688	0.207959	0.068698	0.181308	0.128804	0.113645
9	Beta	0.334427	0.039581	0.055274	0.221698	-0.15425	0.054491	0.065662	0.306002	1	0.264336	0.185007	0.167238	0.32991	-0.04368	0.232848	0.069703	0.155373	0.031436	0.109612	0.189625
10	Natuurkunde	0.098747	0.331823	0.366626	0.287299	0.190545	0.303334	0.326078	0.154016	0.264336	1	0.742402	0.649684	0.1226	0.352481	0.308547	0.442929	0.485344	0.276756	0.345515	0.492555
11	Wiskunde	0.115935	0.303732	0.306828	0.117195	0.221538	0.189715	0.257293	0.220728	0.185007	0.742402	1	0.712563	0.022392	0.536477	0.377908	0.550899	0.499206	0.426188	0.332103	0.565189
12	Scheikunde	0.129065	0.33012	0.238261	0.22614	0.184023	0.304797	0.202639	0.207417	0.167238	0.649684	0.712563	1	0.184871	0.527667	0.401127	0.445291	0.506149	0.320772	0.436023	0.488489
13	Gamma	0.284435	0.167637	0.102456	0.311941	0.112682	0.346771	0.232618	0.183841	0.32991	0.1226	0.022392	0.184871	1	0.145883	0.372006	0.215889	0.243773	0.262441	0.267705	0.260744
14	Aardrijkskunde	0.15467	0.412565	0.330775	0.300932	0.320978	0.403168	0.249563	0.124693	-0.04368	0.352481	0.536477	0.527667	0.145883	1	0.336608	0.468895	0.409878	0.412768	0.479221	0.340853
15	Geschiedenis	0.111164	0.309376	0.295477	0.282447	0.295429	0.280648	0.219026	0.381688	0.232848	0.308547	0.377908	0.401127	0.372006	0.336608	1	0.398083	0.17495	0.430086	0.333529	0.37553
16	Maatschappijleer	0.218096	0.327912	0.321716	0.19498	0.317433	0.395868	0.248966	0.207959	0.069703	0.442929	0.550899	0.445291	0.215889	0.468895	0.398083	1	0.394418	0.512581	0.486159	0.416699
17	Economie	0.283263	0.292704	0.171882	0.100167	0.16654	0.21771	0.198302	0.068698	0.155373	0.485344	0.499206	0.506149	0.243773	0.409878	0.17495	0.394418	1	0.285192	0.491504	0.425722
18	Levensbeschouwing	0.106387	0.369122	0.21733	0.207341	0.281434	0.363707	0.298518	0.181308	0.031436	0.276756	0.426188	0.320772	0.262441	0.412768	0.430086	0.512581	0.285192	1	0.476143	0.642039
19	Culture kunstzinnige vorming	0.164334	0.409672	0.343743	0.376603	0.293588	0.499856	0.438952	0.128804	0.109612	0.345515	0.332103	0.436023	0.267705	0.479221	0.333529	0.486159	0.491504	0.476143	1	0.445867
20	Filosofie	0.179834	0.369324	0.184443	0.212749	0.236692	0.296231	0.294298	0.113645	0.189625	0.492555	0.565189	0.488489	0.260744	0.340853	0.37553	0.416699	0.425722	0.642039	0.445867	1

Figure D.1: The similarity matrix for middle school courses of the FastText model



numme	naam	Accord	Dwarsf	Blokflu	Hobo	Fagot	Saxofo	Klarine	Trombo	Trompe	Tuba	Chordo	Harp	Snaarin	Citer	Clavine	Piano
1	Accordeon	1	0.619745	0.65104	0.468789	0.379173	0.671178	0.484803	0.568763	0.539188	0.202454	0.584905	0.168373	0.726298	0.261421	0.366151	0.340788
2	Dwarsfluit	0.619745	1	0.77763	0.281953	0.625662	0.746114	0.830056	0.712038	0.629961	0.302856	0.487724	0.141196	0.657912	0.37895	0.532438	0.323434
3	Blokfluit	0.65104	0.77763	1	0.355277	0.588842	0.697889	0.727783	0.653759	0.587942	0.330373	0.643398	0.156487	0.723041	0.353388	0.480816	0.307685
4	Hobo	0.468789	0.281953	0.355277	1	0.232372	0.331155	0.201058	0.196437	0.221324	0.289287	0.412229	0.339605	0.442885	0.229089	0.17712	0.287428
5	Fagot	0.379173	0.625662	0.588842	0.232372	1	0.565879	0.710414	0.630333	0.617335	0.27442	0.484741	0.183142	0.406129	0.353681	0.528482	0.258163
6	Saxofoon	0.671178	0.746114	0.697889	0.331155	0.565879	1	0.716001	0.682345	0.681755	0.312594	0.67003	0.091615	0.686039	0.357342	0.49396	0.305626
7	Klarinet	0.484803	0.830056	0.727783	0.201058	0.710414	0.716001	1	0.749342	0.723409	0.236359	0.501745	0.167244	0.569715	0.300595	0.518847	0.333194
8	Trombone	0.568763	0.712038	0.653759	0.196437	0.630333	0.682345	0.749342	1	0.720815	0.172216	0.526474	0.216444	0.588786	0.359765	0.533504	0.227798
9	Trompet	0.539188	0.629961	0.587942	0.221324	0.617335	0.681755	0.723409	0.720815	1	0.265075	0.509012	0.193849	0.551319	0.286937	0.350849	0.178564
10	Tuba	0.202454	0.302856	0.330373	0.289287	0.27442	0.312594	0.236359	0.172216	0.265075	1	0.335283	-0.04399	0.235694	0.20059	0.358617	0.431142
11	Chordofoon	0.584905	0.487724	0.643398	0.412229	0.484741	0.67003	0.501745	0.526474	0.509012	0.335283	1	0.174471	0.662265	0.336898	0.435593	0.352362
12	Harp	0.168373	0.141196	0.156487	0.339605	0.183142	0.091615	0.167244	0.216444	0.193849	-0.04399	0.174471	1	0.230117	0.163735	0.116045	0.039012
13	Snaarinstrument	0.726298	0.657912	0.723041	0.442885	0.406129	0.686039	0.569715	0.588786	0.551319	0.235694	0.662265	0.230117	1	0.334293	0.484098	0.304887
14	Citer	0.261421	0.37895	0.353388	0.229089	0.353681	0.357342	0.300595	0.359765	0.286937	0.20059	0.336898	0.163735	0.334293	1	0.345205	0.319706
15	Clavinet	0.366151	0.532438	0.480816	0.17712	0.528482	0.49396	0.518847	0.533504	0.350849	0.358617	0.435593	0.116045	0.484098	0.345205	1	0.345422
16	Piano	0.340788	0.323434	0.307685	0.287428	0.258163	0.305626	0.333194	0.227798	0.178564	0.431142	0.352362	0.039012	0.304887	0.319706	0.345422	1

Figure D.2: The similarity matrix for instruments of the FastText model

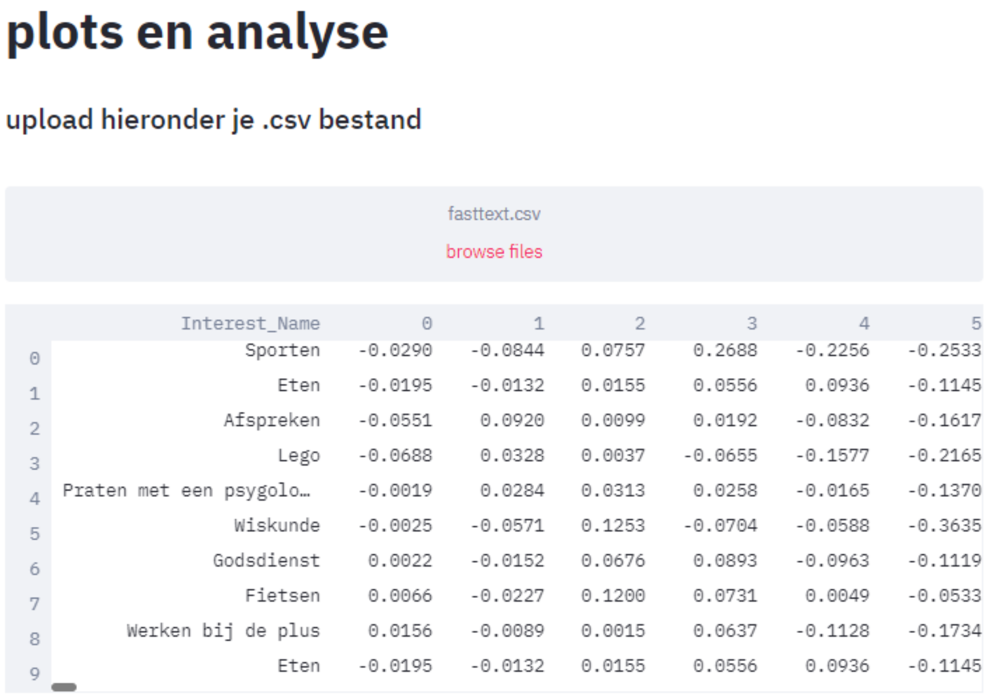


numme	naam	Alpinis	Americ	Autosp	Backga	Badmir	Balspor	Bandy	Basejur	Basketl	Beachv	Bergsp	Biatlon	Biljart	BMX
1	Alpinisme	1	0.132457	0.20465	0.251033	0.341019	0.460922	0.282378	0.410186	0.358743	0.290519	0.662321	0.380116	0.432528	0.05807
2	American football	0.132457	1	0.299881	0.234181	0.466933	0.275528	0.247244	0.179416	0.395487	0.364703	0.267957	0.266085	0.31793	-0.06461
3	Autosport	0.20465	0.299881	1	0.272793	0.325026	0.489181	0.063432	0.372706	0.432376	0.46232	0.44483	0.511653	0.321329	-0.07465
4	Backgammon	0.251033	0.234181	0.272793	1	0.32136	0.446271	0.281968	0.39747	0.370809	0.299531	0.392047	0.516404	0.509209	0.182263
5	Badminton	0.341019	0.466933	0.325026	0.32136	1	0.314153	0.404256	0.359801	0.517664	0.435526	0.39756	0.398615	0.492651	0.080109
6	Balsport	0.460922	0.275528	0.489181	0.446271	0.314153	1	0.246341	0.407854	0.538105	0.461412	0.677797	0.572966	0.514764	-0.02077
7	Bandy	0.282378	0.247244	0.063432	0.281968	0.404256	0.246341	1	0.443242	0.326675	0.098618	0.219906	0.298167	0.243199	0.182025
8	Basejumpen	0.410186	0.179416	0.372706	0.39747	0.359801	0.407854	0.443242	1	0.385886	0.39185	0.482469	0.483999	0.525338	0.145892
9	Basketbal	0.358743	0.395487	0.432376	0.370809	0.517664	0.538105	0.326675	0.385886	1	0.628772	0.375497	0.505614	0.406982	0.0798
10	Beachvolleybal	0.290519	0.364703	0.46232	0.299531	0.435526	0.461412	0.098618	0.39185	0.628772	1	0.259699	0.526135	0.36782	0.02283
11	Bergsport	0.662321	0.267957	0.44483	0.392047	0.39756	0.677797	0.219906	0.482469	0.375497	0.259699	1	0.445207	0.499698	0.015043
12	Biatlon	0.380116	0.266085	0.511653	0.516404	0.398615	0.572966	0.298167	0.483999	0.505614	0.526135	0.445207	1	0.558709	0.138792
13	Biljart	0.432528	0.31793	0.321329	0.509209	0.492651	0.514764	0.243199	0.525338	0.406982	0.36782	0.499698	0.558709	1	0.154016
14	BMX	0.05807	-0.06461	-0.07465	0.182263	0.080109	-0.02077	0.182025	0.145892	0.0798	0.02283	0.015043	0.138792	0.154016	1

Figure D.3: The similarity matrix for sports of the FastText model

Appendix E

This appendix provides screenshots of the tool. The full code of the tool plus embedded interests are provided in the repository.



plots en analyse

upload hieronder je .csv bestand

	Interest_Name	0	1	2	3	4	5
0	Sporten	-0.0290	-0.0844	0.0757	0.2688	-0.2256	-0.2533
1	Eten	-0.0195	-0.0132	0.0155	0.0556	0.0936	-0.1145
2	Afspreken	-0.0551	0.0920	0.0099	0.0192	-0.0832	-0.1617
3	Lego	-0.0688	0.0328	0.0037	-0.0655	-0.1577	-0.2165
4	Praten met een psygolo…	-0.0019	0.0284	0.0313	0.0258	-0.0165	-0.1370
5	Wiskunde	-0.0025	-0.0571	0.1253	-0.0704	-0.0588	-0.3635
6	Godsdienst	0.0022	-0.0152	0.0676	0.0893	-0.0963	-0.1119
7	Fietsen	0.0066	-0.0227	0.1200	0.0731	0.0049	-0.0533
8	Werken bij de plus	0.0156	-0.0089	0.0015	0.0637	-0.1128	-0.1734
9	Eten	-0.0195	-0.0132	0.0155	0.0556	0.0936	-0.1145

Figure E.1: The file upload module of the tool

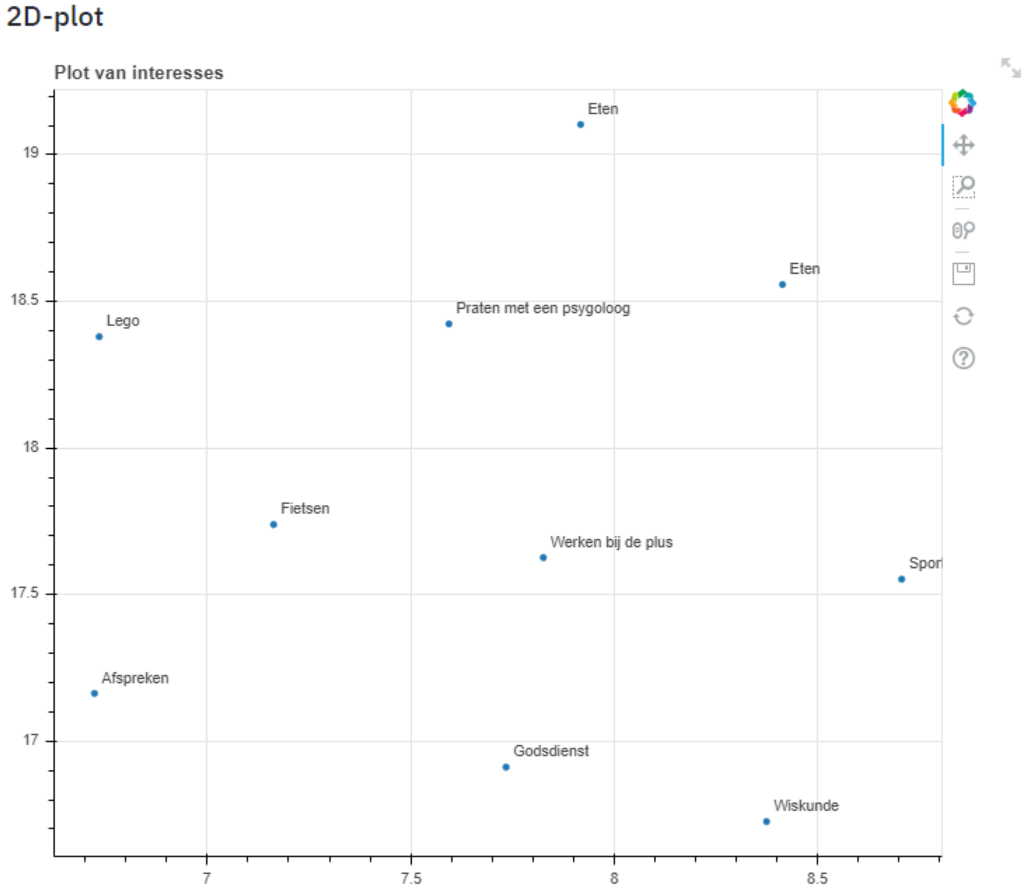
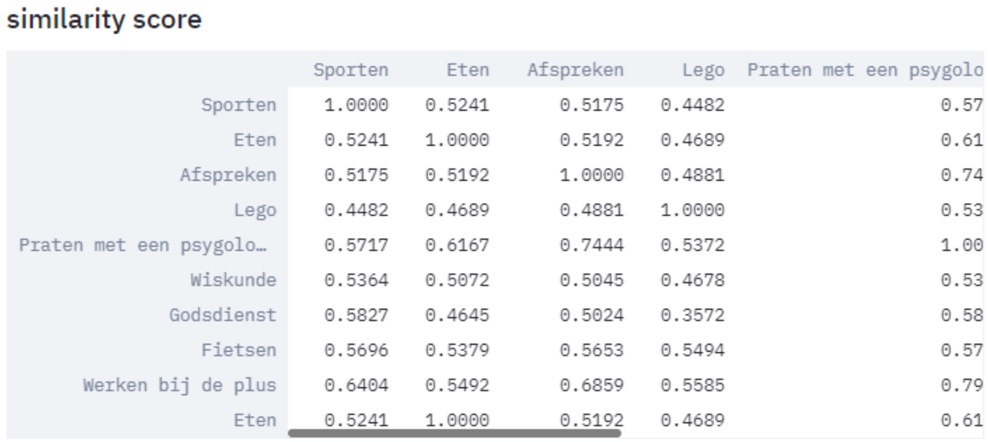


Figure E.2: The plotting module of the tool



similarity score

	Sporten	Eten	Afspreken	Lego	Praten met een psygolo
Sporten	1.0000	0.5241	0.5175	0.4482	0.57
Eten	0.5241	1.0000	0.5192	0.4689	0.61
Afspreken	0.5175	0.5192	1.0000	0.4881	0.74
Lego	0.4482	0.4689	0.4881	1.0000	0.53
Praten met een psygolo…	0.5717	0.6167	0.7444	0.5372	1.00
Wiskunde	0.5364	0.5072	0.5045	0.4678	0.53
Godsdienst	0.5827	0.4645	0.5024	0.3572	0.58
Fietsen	0.5696	0.5379	0.5653	0.5494	0.57
Werken bij de plus	0.6404	0.5492	0.6859	0.5585	0.79
Eten	0.5241	1.0000	0.5192	0.4689	0.61

Figure E.3: The similarity score module of the tool

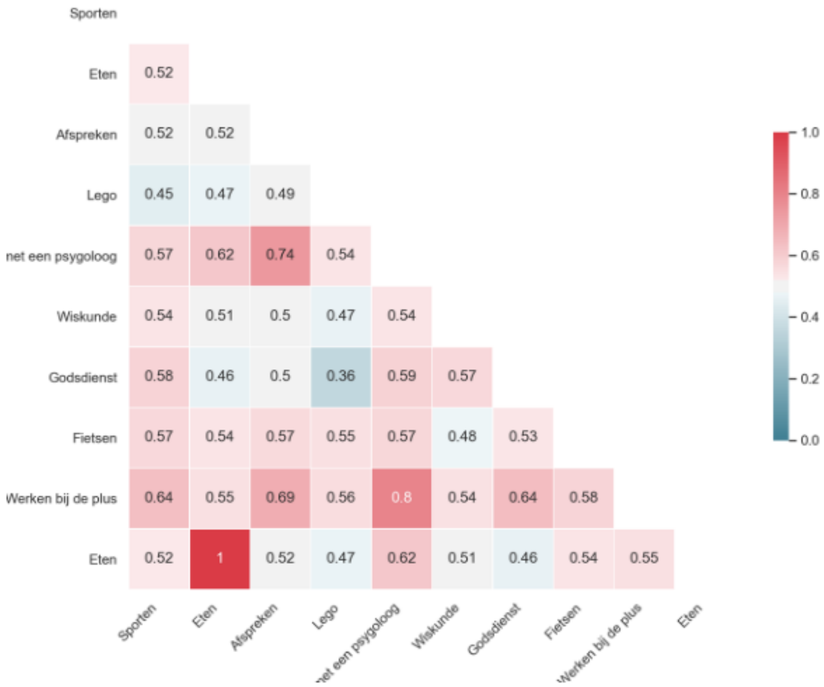


Figure E.4: The heat map module of the tool

Appendix F

A Deep Learning Approach to Interest Analysis

Thomas van der Meer
Utrecht University
Affiliation / Address line 2
Affiliation / Address line 3
t.vandermeer@uu.nl

Dr. Marco Spruit
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

The analysis of interests from young adolescents in the form of short, colloquial Dutch text is a challenging task for pre-trained neural networks. By quantitative and qualitative tests, four pre-trained language models, including three language model fine-tuned models, are compared and contrasted on the Dutch language. The goal is to effectively analyse Dutch interests on semantical relations. Firstly, the models are evaluated on quantitative tests. The Qualitative tests consist of triplets, clustering and structure analysis, focusing on the output from the embedding layer of the models.

1 Introduction

The individual interests of people are unique and develop from situations throughout life (Akkerman and Bakker, 2019). In order to understand the interest development, daily activities of people are tracked to gain insight. The currently running research project titled “Lost in Transition: Multiple Interests in Contexts of Education, Leisure and Work” gathers this data with the goal to find out how different interests relate to each other and how these interests develop over time. This is done through an experience sampling method (ESM) data collection process, where events throughout the day are recorded by the user themselves.

The nature of the data are short texts, written in colloquial Dutch, containing possible slang, misspellings and other contaminations. These properties provide an extra challenge. The task at hand is to analyse the data effectively through automated methods in order to ultimately map interest development of people over time.

The research landscape of natural language processing (NLP) has radically changed over the last decade. The ability to train word representation models with vast amounts of natural language (Mikolov et al., 2013b), has been a catalyst to a wide range of new techniques. Using unsupervised training of a neural network on large datasets, a sense of syntactical, semantical and contextual awareness can be found in the word representations (Peters et al., 2018). Language model fine-tuning (LMFT) (Howard and Ruder, 2018) is important to capture the idiosyncrasies of the target corpus.

In order to tackle the problem of analysing short, colloquial Dutch text, pre-trained language models are used to effectively process the data into vector representations. These representations are used to uncover relations between the interests and used to show the interest development for young adolescents over time.

The scientific contribution of this paper is the in-depth analysis of the output embeddings of various Dutch NLP models. The qualitative tasks are an addition for exhaustive exploration of the vector representations of language models. Lastly, a comparison between different language model architectures is made that uncovers motivations the relational structures are based on.

This paper is structured as follows. Section 2 introduces the development of NLP models and architectures over the last years. There is a special focus for native Dutch NLP models. Section 3 will focus on the method used in order to quantitatively and qualitatively vet the selected NLP models. The results of the experiment are laid out in section 4. The paper is concluded in section 5.

2 Related work

In this section, the evolution of language models over the last decade are discussed (2.1). This is followed by the pre-training of those model architectures on Dutch data and the models that are conceived from this (2.2).

2.1 Language models

The goal of language modeling, central to NLP, as given by Jozefowicz et al. (2016) is ”to learn a probability distribution over sequences of symbols pertaining to a language.” This has historically been done through different methods, such as a parametric model, count-based and since the current decade more through neural networks. To put into context, a five-gram (probability over five words) model from 1995 has been a strong baseline that has been competitive with neural network approaches (Jozefowicz et al., 2016).

The enabler of the pre-trained neural network language models that are seen today, is the research of Mikolov, Chen et al. (2013a). The ability to pre-train, unsupervised, a neural network on text was a breakthrough. The neural network architectures were already invented (Rumelhart et al., 1986; Hochreiter and Schmidhuber, 1997), the training methodology now enabled training on billion word sized corpora with unlimited vocabulary sizes (Mikolov et al., 2013a).

Since the inception of this efficient training method, the first pre-trained language model through this method was made available, named Word2Vec (Mikolov et al., 2013b). The language model training made use of the Skip-gram architecture to learn word embeddings, namely using the input (word) to predict the surrounding words (window size). This method resulted in a model that is able to capture semantic relationships (Mikolov et al., 2013b). Linzen (2016) however, is more critical and proves that this is not completely true, especially for analogies.

Extensions on this method has become apparent, providing new pre-trained models with methods such as out-of-vocabulary (OOV) capabilities (Bojanowski et al., 2017). OOV words do not get a vector representation from the Word2Vec model (Mikolov et al., 2013b), because there is no information remembered about this word. The FastText model of Bojanowski et al. (2017) solves this by using subword information to create an embedding. Bojanowski et al. (2017) show that OOV representations work fairly well, showing relatively correct word similarities built from subword information.

Model architectures also have evolved drastically over the last decade. Where the models above use recurrent neural networks (RNN) (Rumelhart et al., 1986), the architectures also evolve over time. To improve upon RNN's on long term dependencies, long term short memory (LSTM) model architectures (Hochreiter and Schmidhuber, 1997) are used in language models (Jozefowicz et al., 2016; Peters et al., 2018). The LSTM architecture implements a 'cell' with gates that can be trained in order to store long term dependencies.

Vaswani et al. (2017) introduce the Transformer architecture, enabling a new wave of language models using this neural network architecture. The key component of the transformer is the attention mechanism. The attention mechanism takes in a longer input consisting of multiple parts, like a sentence, and captures the input. To capture a whole sentence in one way is not enough, so multiple attention 'heads' are used to catch different aspects of the sentence. These inputs are captured by the encoder from the transformer. The decoder uses the information from the multi-head attention to create a prediction. To relate this to pre-trained language models, The GPT models from Openai use the transformer architecture (Radford et al., 2018; Radford and Salimans, 2018).

Another improvement upon the training of neural network, is bi-directionality. When having a sequential input using the previous context to predict the next, this can also be done using the future context predicting the input before. This idea is formalized by Peters et al. (2017). The biLM, concatenates the prediction of a forward- and backward language model in order to predict the correct output. This notion of bi-directionality is also important for at that moment future state of the art (SOTA) language models like BERT (Devlin et al., 2018).

Where both the GPT and BERT models use the transformer architecture of Vaswani et al. (2017), BERT differs in pre-training procedure. Using masked language modeling (MLM) the BERT model learns bidirectional representations. MLM uses a sentence as input, but fifteen percent of the words in the sentence are masked with a token. The model has to predict what should be on the spot of the token (Devlin et al., 2018).

Finally, RoBERTa (Liu et al., 2019) is an extension on the BERT model. liu et al. (2019) found in their replication study that the BERT model from Devlin et al. (2018) was 'significantly undertrained' and therefore, by choosing different hyperparameters and more data, the RoBERTa model is created (Liu et al., 2019).

The final model that is discussed in this section is the Flair model (Akbik et al., 2018). The Flair model is unlike the word- (Mikolov et al., 2013b; Bojanowski et al., 2017) and sentence (Devlin et al., 2018; Liu et al., 2019) models, a model based on the sequence of characters. With the use of a bidirectional architecture, the model learns contextualized representations with SOTA performance (Akbik et al., 2018).

2.2 Dutch language models

The models described above, also have a Dutch implementation. This ranges from a FastText implementation trained on only the Dutch Wikipedia, to robBERT, a RoBERTa implementation on a collection of corpora, a total of 39GB. An overview of the models with their respective input type, architecture and corpora trained on, is given in table 1.

When it comes to the models based on the transformer architecture, there are multiple Dutch models available. to For Dutch tasks, de Vries et al. (2019) showed that the BERTje model outperformed the multilingual implementation of Devlin et al. (2018) BERTje was again outperformed by RobBERT (Delobelle et al., 2020), the on Dutch corpora trained version of RoBERTa (Liu et al., 2019).

3 Method

In this section, the method for evaluating the Dutch pre-trained language models is described. The model evaluation is done quantitatively (section 3.1) and qualitatively (section 3.2). The quantitative method consists out of two parts, namely the CoNLL 2002 named entity recognition (NER) task (3.1.1) and the Dutch book review database (DBRD) sentiment analysis task (3.1.2). The qualitative method consists out of three tests. The triplets test (3.2.1), clustering test (3.2.2) and the structure test (3.2.3).

3.1 Quantitative analysis

To create an overall assessment of the different Dutch NLP models, two baseline tasks are identified, trained on and evaluated. For the quantitative tasks, four models are used, namely the FastText, Flair, BERTje and robBERT model. Note that for the FastText and Flair model, the models trained on Dutch are used (self-evidently also trained on Dutch for the BERTje and robBERT).

3.1.1 CoNLL 2002 NER task

The first baseline task is the CoNLL 2002 NER task (Tjong Kim Sang, 2002) where the model has to recognise four entities. The entities are person, location, organization and miscellaneous. The dataset is provided through the Flair framework[1], including the code to run this task for both the Flair and FastText embeddings. The training runs for 50 epochs, a full cycle over all the training data, while the other hyperparameters are kept standard. The transformer models are trained for the same amount of epochs in the Huggingface transformers framework[2]. The metric used for evaluating the models is the F1-score.

3.1.2 DBRD sentiment analysis task

The second task is a sentiment analysis binary classification task on the 110kDBRD[3]. The dataset consists out of 110 thousand book reviews scraped from Dutch book review website Hebban[4]. A balanced training subset of more than twenty thousand reviews is trained, followed by a ten percent test set to evaluate the model. After training for four epochs, the model is tested on the test set and the Matthews Corrlation Coefficient (MCC) score is calculated.

3.2 Qualitative analysis

The goal of the qualitative tests is to see how the models interpret interest data. The focus lies on the outputs of the embedding layer of the models. These outputs are obtained through the Flair framework for the FastText and Flair models and the Huggingface Transformers framework for BERTje and robBERT. The embeddings returned are not the same dimensions for the models, ranging from 300 to 768.

The qualitative analysis is done for seven models. The three extra models are LMFT models of the Flair, BERTje and robBERT models. The models are fine-tuned on Dutch Reddit data from subreddit /r/thenetherlands[5]. The dataset consists of scraped comments ,resulting in a set of more than 70.000 comments. The Flair model is fine-tuned through the Flair framework and BERTje

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

[2] https://github.com/huggingface/transformers

[3] https://github.com/benjaminvdb/110kDBRD

[4] https://www.hebban.nl/

[5] https://reddit.com/r/thenetherlands

Model name	Input type	Architecture	Trained on
FastText	Word-based	RNN	Wikipedia/Common Crawl
Flair	Character-based	Forward and backward RNN	OPUS
BERTje	Sentence-based	Transformer	Books, TwNC, SoNaR-500, Wikipedia, Web News
RobBERT	Sentence-based	Transformer	OSCAR

Table 1: An overview of the input type, architecture and pre-training datasets

and robBERT through the Hugginface transformers framework.

3.2.1 Triplets test

The goal of the first experiment is to see if the model interprets interests in the same way experts do. For the first test, the method of the experiment is as follows. The data consists out of 100 samples. one sample consists out of three interests, forming a triplet. The experts rank the similarity of the interests from most similar to least similar. The experts have created a 7-dimensional interest relation classification form (table 5 in appendix) to score the relation between interests on different aspects. The triplets are created by the experts separately and later on discussed until agreement of the scores.

The model has to do the same. The three interests are embedded by the model and the output from the embedding layer is returned. To see if the model understands the different interests, the relative similarity of the embeddings is calculated, using cosine similarity[6]. The closer the cosine similarity score is to one, the more similar the interest are.

Finally, the classification of the models is compared to the classification of the experts. To measure the ranking of the relative interests, there are three groups. Interest classification done totally correct, so all interests correlations are ranked correctly. Secondly, ranking where only one interest correlation ranking is correct of the three. Finally there are the rankings that are totally incorrect.

3.2.2 Clustering test

To move to the clustering test, the goal changes to a more broader approach to interest analysis and how the interests relate. Where the triplets test looked and individual samples where three interests were inspected closely, this clustering test looks at the big picture where the structure of hundreds of interests are investigated.

The second test is a blind test where experts get a two-dimensional plot of all the unique interests that occur in the data gathering period for one school. In order to bring back the highly-dimensional data that is returned from the models to a two-dimensional plot, UMAP is used (McInnes et al., 2018). The plots are filled with annotated data points. Furthermore, the researchers are also provided with an unsupervised clustered set of interest, using HBDSCAN (McInnes et al., 2017).

While investigating the plots, the researchers are given the task to take into account the distribution of the interests, looking for a structure of related interests grouped together, recognising overarching practices or the odd one out. Using the different plots and comparing the different structuring, the desire is to better understand the semantic connections the models make.

Additionally, the researchers are also provided a spreadsheet with a list of the different clusters and outliers. The researchers have to name the clusters and see if the clusters have interests that do not belong to the cluster. Furthermore, the experts describe what they think the factors are that the cluster is based on. Lastly, the list of outliers is inspected in order to see if the outliers do not fit into one of the clusters. The naming of the clusters is conducted in order to indicate the coherence of the clusters and the possible outliers that can adhere to this cluster.

3.2.3 Structure test

The third and last test is investigation of the ability to categorize words correctly. For this categorisation, the Dutch Wikipedia category tree[7]is used. The categories are musical instruments, sports and middle school course. In order to bring back the number of sports and instruments, a selection was

[6] https://scikit-learn.org/stable/modules/metrics.html#cosine-similarity

made to make sure the items are correct and more generally known, so that the item is included in the vocabulary of the model.

The models have as input the lists of sports, instruments and courses. The correlation score of these vector representations are computed and a matrix is created to have a comparison to all. The researchers take a sample that is consistent over all models to check how the instruments and sports are correlated. For the school courses, all items are checked.

4 Results

The results in this section are described in the same order as the method (section 3) is. The full results and spreadsheets and scores are made available through the following repository[8].

4.1 Quantitative results

4.1.1 CoNLL 2002 NER task

The full results are in table 2. The scores are expressed in F1-score, accompanied with precision, recall and the training loss of the model. The order is based on the F1-score from best to worst.

The F1-score has a scale from 0 to 1 and consists out of the harmonic mean of precision and recall. The F1-scores all fall between a range of 0.1 difference. For the precision, this window is only 0.01858. Recall has more spread between low .90 and high .78. The lowest training loss is recorded at 0.0881 and the highest training loss is 0.7698.

The models are close in performance on this task, especially in the precision metric, where the models score between .03 of each other. The difference is made in the recall of the models, where the FastText is scoring considerably lower than the other models. RobBERT scores well in both categories but just a little less than the Flair model does. On top is the BERTje model with a .9003 score.

Another metric that stands out is the training loss. The training loss is the error the function has on the training set while training. The lower the training score, the better the model can fulfil the task on the training set. If the model can generalize well and does not overfit on the training set, this will result in a high score on the test set. The FastText model seems to not fit to the test set well and it shows in the training loss (0.7698). It does not come close to the training losses of the other 3 models. What stands out is that with a higher training loss, 0.1317 to 0.1091, Flair outperforms RobBERT on the test set. It cannot be concluded that there is overfitting but it seems that RobBERT does not generalize well from the training- to test set.

4.1.2 DBRD sentiment analysis task

The scores are expressed in the MCC score. The MCC metric has a scale of -1 to 1. A score of 1 is a perfect score, so all the scores are equal to the test set truth. A score of 0 is equal to random chance. A score of -1 is total disagreement. Table 3 shows the results of the different models on the DBRD task.

Flair and FastText do have the hardest time to perform in this task, both scoring considerably lower than the Transformer-based models. The big difference between the quantitative tests is the nature of the data, where the task consists of understanding the sentiment of full sentences. If there is an interplay between words, one could argue that transformer-based models have an easier time understanding through their attention mechanisms than through a RNN for both the other models. As a general remark, the difference in training method can have an influence on the final scores.

Model name	MCC-score
BERTje	0.85072
RobBERT	0.76551
FastText	0.60433
Flair	0.54329

Table 3: Results from the DBRD task

4.2 Qualitative results

4.2.1 triplets test

Table 4 shows the results of the different models on the triplets test. The columns show the number of correct, partially correct and incorrect rankings compared to the expert evaluation. The models followed by the letter FT are the fine-tuned models. The RobBERT model that is fine-tuned on Reddit texts is performing the best with 29 of the 100 correct rankings. FastText is second, while have less rankings correct it has more partially correct answers and less incorrect answers. Flair fine-tuned, Flair and RobBERT are all close together, with Flair and RobBERT scoring the exact same score. Both the BERTje models have the lowest number of correct scores of all the models.

[8] https://git.science.uu.nl/tvdermeer/thesis

Model name	F1-score	Precision	Recall	Training loss
BERTje	0.90309	0.89961	0.90660	0.0881
Flair	0.87603	0.88088	0.87195	0.1317
RobBERT	0.86372	0.86216	0.86528	0.1091
FastText	0.82795	0.88103	0.78928	0.7698

Table 2: Results from the CoNLL task

4.2.2 Language model fine tuning

To compare and contrast the LMFT models against their standard model counterpart, the fine-tuning seems to have different effects on the different models. For the robBERT model, the model score improves. The number of incorrect answers stays almost the same, while the partially correct answers are changed into fully correct answers. A notable difference when looking at the correlation scores distribution of the fine-tuned and non-fine-tuned model is that the fine-tuned model has a greater spread. When looking at the other transformer model, BERT, this is not case the when it is fine-tuned. The influence of the LMFT is not as apparent. There are more correctly ranked triplets, but there is also an increase in incorrect triplets.

For the Flair models, the scores get better by fine-tuning, but only minimally (one extra correct, three less incorrect). However, when looking at the incorrect scores for both the models, only twelve are similar (of the 31). This means that roughly twenty incorrect answers do not overlap. This is the same for the correct answers, where only ten are similar of the 23. This phenomenon can be the product of the fine-tuning heavily affecting the model where there is previous information in the model is lost.

4.2.3 Clustering test

The general conclusion of the experts was that the results were not as expected. The experts evaluated the clusters as superficial and morphological, not given the substantive meaning of the embeddings. Clusters are made from similarities in verb/noun, abbreviation and matching words, not the meaning.

The models were all suffering from the same shortcomings. Firstly, the degree of how the models react on parts of words that are equal, for example the use of social-. To extend the example, Social media and Social worker are two very different things. Secondly, the models seem to react on characters that do not often occur (figure 1 in appendix).

The question was raised if the plotting and clustering methods do influence the embeddings so much that the interpretation of the embedding is not ‘pure’ anymore. To expand on this, do the dimension reduction technique in UMAP and the clustering algorithm HDBSCAN take away the subtleties from the vector representations? Closer inspection of the similarity matrices made for the models to inspect discrepancies between the models and the plots, shows that there was no such thing found. The resulting two-dimensions from the vectors of the embedding layer was not affected significantly by the dimension reduction techniques and the clustering method was seemingly working the way it should.

To investigate the raw embeddings more, the deductive interest analysis takes a look at the similarity scores between items that are widely understood as similar in a sense and therefore can verify the quality of the embeddings. The underlying problem here is the degree of meaning the model has of the language. The interpretation of the researchers is that the model focuses on a lot of the morphology of words, but not necessary in semantics of words.

4.2.4 Structure test

The middle school courses, broken down in alpha, beta and gamma, worked generally well for all the models. Most of the models could distinguish the languages very well, except for flair FT that had difficulties with Latin. BERT could not correlate French high with the other languages and FastText could not work well with German. The beta courses also share the same word part in ‘-kunde’. Therefore this score is less significant because the high correlation can also be based on consisting of the same word part. For the gamma courses, this was the hardest of the middle school course categories. Multiple models had difficulties to distinguish the courses like biology, geography and economy from the beta courses. Math was scoring high on the gamma courses for BERTje.

Model name	Correct	Partially correct	Incorrect
RobBERT (FT)	29	39	32
FastText	27	47	26
Flair (FT)	24	45	31
Flair	23	43	34
RobBERT	23	43	34
BERTje (FT)	20	46	34
BERTje	17	54	29

Table 4: Results from the triplet test. (FT) Stands for fine-tuned.

For RobBERT, history was correlating the highest with math. For multiple models, Philosophy was correlating high with beta courses.

Unfortunately, the good performance of the models on the courses is not continued in the sports categories. The sample consists of American football, badminton, bridge, hockey and bergsport. For American football, the models could not understand well that it was a ball sport. The Flair model was correlating English words high with American football. BERTje correlated words with 'bal' in it. For badminton, a racket sport, tennis was nowhere to be found in all the models. FastText correlated 'rolstoel' high with badminton. FastText was working relatively well on the other sports in bridge and hockey. The other models could not interpret bridge as a mind sport. BERTje is correlating hockey the highest with football and tennis, probably due to the popularity of the sports. The last sport, 'bergsport' was only correlating with words that also had sport in it. This can be caused by not being in the vocabulary and falling back on word parts.

The last category are the instruments, that are judged on the following instruments; zither, trumpet, recorder and synthesizer. For all the instruments, the models were performing poorly. Even a trumpet, a widely known instrument, does not work well and is not getting close to other wind instruments. Only FastText could disambiguate some instruments and create some good correlations.

5 Conclusion

Bertje (de Vries et al., 2019) is the best model for the quantitative tasks (4.1). The model fine-tuning on a training set seems to work effectively and the model does not seem to be overfitted. Unfortunately, the results from the models on the quantitative tasks, are not carried over on the qualitative tests.

As the results from the qualitative analysis (4.2) portray best, none of the models possess output embeddings that enable Dutch interest analysis to be done unsupervised. The models output embeddings from the embedding layer do not show the semantic relations that were expected to be found in the triplets (4.2.1), clustering (4.2.3) and structure 4.2.4 tests. As shown in the clustering and structure tests, the model similarity score are mainly driven by the morphological similarity found in the interests and structures.

Applying LMFT to the models, seem to effectively change the outputs from the embedding, but is not always an improvement. The model that seems to react the best, is the robBERTa (Delobelle et al., 2020) model.

5.1 Future research

Where this research uses the vector representations for measuring the relation to other words, creating clusters and plotting overall distances to other representations, an unexplored option is to create a downstream task for this. For example, the clustering method can be generalised to a multi-classification problem, where the models are presented with a training set in order to learn the predefined classes (clusters) of interests.

Furthermore, the models and data could be enriched. From a data-centered approach, the interest fields can be enhanced. There is extra data available that describe the interests in more detail. These fields can be added as a whole to the input but a more delicate approach would be using only certain parts of the sentence that is interesting. This can be done through a NER system that selects certain named entities to add to the input.

Secondly, the model enrichment can be done through knowledge infusion. This can be done in multiple ways, through fully training a neural network on knowledge graphs, such as ERNIE (Zhang et al., 2019). Secondly, this can be done through taking a pre-trained model and retraining it, for example KnowbERT (Peters et al., 2019). The last option is training a smaller neural network on knowledge data and fusing this together with a pre-trained model, therefore not having to change the pre-trained model. This method is called K-adapter (Wang et al., 2020).

Acknowledgements

I would like to thank both my supervisors, Marco Spruit and Sanne Akkerman, for their guidance through these invigorating months of working on my thesis. You both have helped me tremendously by always thinking critically and put immeasurable amounts of energy in the theoretical part of this work. Furthermore, I would like to thank Jael Draijer and Joris Beek for their help with the analysis of the qualitative reviews, being there as sounding board and advisors, and most importantly, the comradery at the office and from home. Lastly, Thank you Pablo Mosteiro for your insights in NLP techniques and your technical perspective on all this.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Sanne F. Akkerman and Arthur Bakker. 2019. Persons pursuing multiple objects of interest in multiple contexts. *European Journal of Psychology of Education*, 34(1):1–24.

Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. 2017. Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*, 5:135–146.

Wietse de Vries, Andreas van Cranenburgh, Arianna Bisazza, Tommaso Caselli, Gertjan van Noord, and Malvina Nissim. 2019. BERTje: A Dutch BERT Model.

Pieter Delobelle, Thomas Winters, and Bettina Berendt. 2020. RobBERT: a Dutch RoBERTa-based Language Model.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. (Mlm).

Sepp Hochreiter and Jürgen Schmidhuber. 1997. LSTM can solve hard long time lag problems. *Advances in Neural Information Processing Systems*, pages 473–479.

Jeremy Howard and Sebastian Ruder. 2018. Universal language model fine-tuning for text classification. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:328–339.

Rafal Jozefowicz, Oriol Vinyals, Mike Schuster, Noam Shazeer, and Yonghui Wu. 2016. Exploring the Limits of Language Modeling.

Tal Linzen. 2016. Issues in evaluating semantic spaces using word analogies. (Figure 2):13–18.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. (1).

Leland McInnes, John Healy, and Steve Astels. 2017. hdbscan: Hierarchical density based clustering. *The Journal of Open Source Software*, 2(11), mar.

Leland McInnes, John Healy, Nathaniel Saul, and Lukas Grossberger. 2018. UMAP: Uniform Manifold Approximation and Projection. *The Journal of Open Source Software*, 3(29):861.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013a. Efficient Estimation of Word Representations in Vector Space. pages 1–12.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013b. Distributed representations ofwords and phrases and their compositionality. *Advances in Neural Information Processing Systems*, pages 1–9.

Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. *ACL 2017 - 55th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:1756–1765.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep Contextualized Word Representations. pages 2227–2237.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge Enhanced Contextual Word Representations. pages 43–54.

Alec Radford and Tim Salimans. 2018. Improving Language Understanding by Generative Pre-Training (transformer in real world). *OpenAI*, pages 1–12.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2018. Language Models are Unsupervised Multitask Learners.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533–536.

Erik F Tjong Kim Sang. 2002. Memory-Based Named Entity Recognition. In *Proceedings of the 6th Conference on Natural Language Learning - Volume 20*, COLING-02, pages 1–4, USA. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017-Decem(Nips):5999–6009.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Jianshu Ji, Guihong Cao, Daxin Jiang, and Ming Zhou. 2020. K-Adapter: Infusing Knowledge into Pre-Trained Models with Adapters.

Zhengyan Zhang, Xu Han, Zhiyuan Liu, Xin Jiang, Maosong Sun, and Qun Liu. 2019. ERNIE: Enhanced Language Representation with Informative Entities. pages 1441–1451.

A Appendix

Name	Explanation
Time, rhythm and regularity	When the interests are time bound, are these comparable to the notion of regularity?
Specificity of knowledge and skills	When there is knowledge and skill needed, are these comparable?
Societal knowledge of phenomenon	When societally known, are the interests the same in the matter of culture and history? Both on a micro- (inside the two interests) and macro level (for a broader audience that share culture and history)
Material comparability	When materials are needed to practice the interest, are these materials comparable?
Geographical comparability	When bound to a physical or digital space, are these comparable?
Social necessity and social nature	When bound to someone else, how comparable are these people?
Link to institutions	When bound to institutions, how comparable are these?

Table 5: The dimensions used by the experts for assessing the relation between two interests

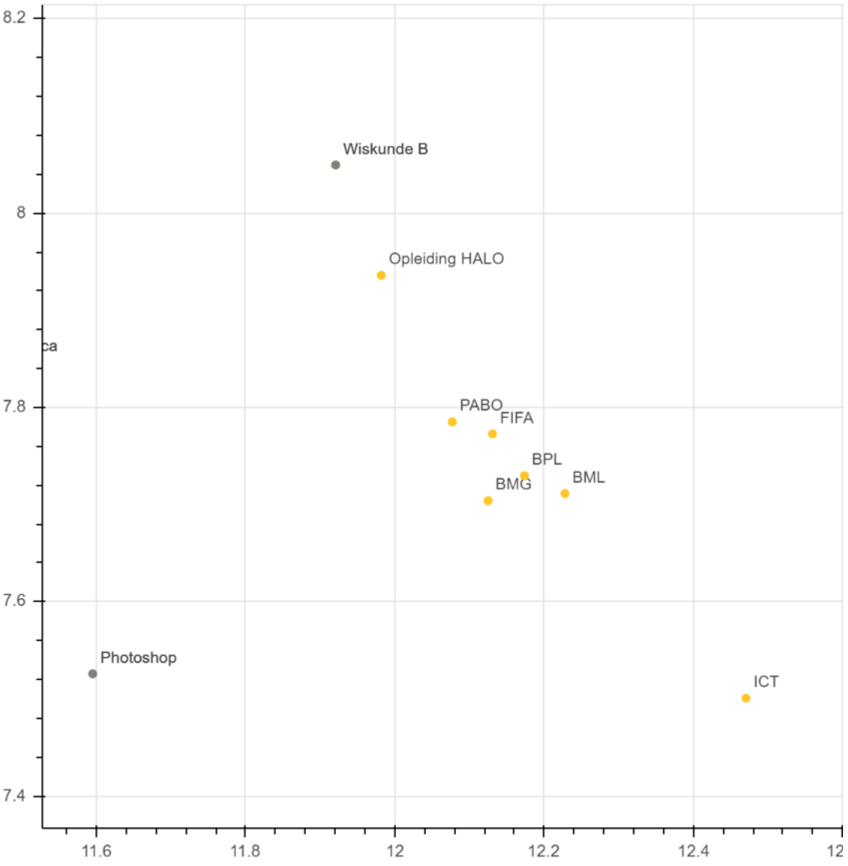


Figure 1: a cluster based on abbreviations