High-dimensional Cluster Analysis of Sentence Embeddings of Verbal 'te'-Infinitival Complement Clauses from Dutch Large Language Models BERTje and RobBERT

Submitted on: **23-02-2025**

Marije Kouyzer
marije.kouyzer@student.uva.nl
University of Amsterdam
Amsterdam, The Netherlands

Jelke Bloem
j.bloem@uva.nl
University of Amsterdam
Amsterdam, The Netherlands

ABSTRACT

The following thesis project will research how linguistic constructions that have the same meaning but a different form are encoded in the output of Dutch BERT models, compared to how linguistic constructions with both a different form and meaning are encoded. The main linguistic constructions that will be researched is the verbal 'te'-infinitival complement clause with and without the complementizer 'om'. These constructions have the same meaning but a different form. A linguistic construction with a different form and the opposite meaning will also be compared, this will be a construction including the word 'niet', meaning 'not'. The Dutch versions of the BERT model that will be used are the BERTje model and the RobBERT model. The data that will be used are sentences with these three different linguistic constructions that will be extracted from the Lassy Groot corpus. The word embeddings will be transformed into sentence embeddings. To compare the cluster coherence of these different linguistic constructions, the following cluster analysis methods will be used: the Dunn Index, the Silhouette Coefficient and the Davies-Bouldin Index. By comparing the cluster coherence of the linguistic constructions with different forms and different meanings, we will learn more about the encoding of form and meaning of linguistic constructions by the BERTje and RobBERT models.

KEYWORDS

Large Language Models, Explainable Artificial Intelligence, Sentence Embeddings, Cluster Analysis, 'te'-Infinitival Complement Clauses, BERTje, RobBERT, Dunn Index, Silhouette Coefficient, Davies-Bouldin Index

GITHUB REPOSITORY

https://github.com/Marije-Kouyzer/Thesis-Data-Science

1 INTRODUCTION

With the rising popularity of the use of Large Language Models (LLMs) in many different contexts, it is more important than ever that we have a better understanding of how they work. These LLMs are generally 'black box' models. This means that researchers, while they understand the inputs and outputs of these models, ultimately do not know how they work internally. They do not know why the models make certain decisions and not others. The growing field that is attempting to learn more about how these models work is called Explainable Artificial Intelligence (XAI). This paper specifically will focus on post-modeling explainability [15], which is the approach where we try to learn more about how existing black box models work internally.

One of the aspects in explainability of LLMs we are interested in is how well linguistic concepts found in human language are encoded, to see how similar it is to the way humans use language. In this paper we will research how well linguistic constructions are encoded by Dutch BERT models and compare this to the encoding of meaning by the same models.

The theory of Construction Grammar defines linguistic constructions as learned pairs of forms and meaning [8]. These can vary from the meaning of individual words to those of complex syntactical patterns. In this paper we will focus on Dutch sentences with a verbal 'te'-infinitival complement clause that have an optional complementizer 'om' [2] [10]. The following Dutch example sentences contain this infinitival complement clause where the complementizer 'om' can be included, but is also grammatical and has the same meaning without this complementizer. The English translation of these sentences is included below each sentence.

(1) *Zij beloven (om) te helpen.*
They promise to help.
(2) *Ik ben vergeten (om) eieren te kopen.*
I forget to buy eggs.
(3) *Jij besluit (om) naar Zuid Amerika te reizen.*
You decide to travel to South America.

To contrast the use of these constructions that only differ in form, another construction will be included as well. This construction includes the word 'niet' (meaning 'not') after the verb. Like the constructions described above, this only adds one word to the sentence, but in this case the meaning of the sentence changes to the opposite meaning. By including this construction we can compare the encoding of form of linguistic constructions to the encoding of meaning.

In this paper, we will analyze the encodings these linguistic constructions in two of the most used Dutch BERT models: BERTje [4] and RobBERT [5]. These are both Dutch versions of the well-known pre-trained Bidirectional Encoder Representations from Transformers (BERT) model [6].

This leads us to the following research questions:

- How does the encoding of linguistic constructions in Large Language Models compare when they differ in form and meaning?

- How well do sentence embeddings from the Dutch Large Language Models BERTje and RobBERT encoding the verbal te-infinitival complement clauses with and without the complementizer 'om' cluster separately?
- Do sentence embeddings from the Dutch Large Language Models BERTje and RobBERT encoding the verbal te-infinitival complement clauses with and without the complementizer 'om' have a higher or lower cluster coherence than the verbal te-infinitival complement clauses without the complementizer 'om' and the same construction including the word 'niet', as measured by the Dunn Index, the Silhouette Coefficient, Davies-Boulding Index?

2 RELATED WORK

The following sections will describe previous research that has been done about linguistic constructions in Large Language Model outputs, word and sentence embeddings, and high-dimensional cluster analysis methods.

2.1 Linguistic Constructions in LLM output

Not a lot of research has been done yet analyzing the encoding of linguistic constructions by Large Language Models, especially for Dutch. Previous work in analyzing linguistic constructions in the output of Large Language Models includes the work by Veenboer and Bloem [21]. They used collostructional analysis. This technique uses correlations between linguistic constructions and individual words in these constructions, called collexemes, to find how strongly they are semantically associated with each other, measuring the collexeme strength. This research looked at the X waiting-to-happen construction (e.g. *an accident waiting to happen*) and the ditransitive construction (e.g. *I gave her the envelope*) in English. The X-waiting to happen construction has one open slot for collexemes, while the ditranstive construction has four. All the different words that these slots are filled with are extracted from a corpus, these are then used to calculate the collexeme strengths between the words and the constructions. This study looked if these different collexeme strengths were found in the output of BERT models, using Masked Language Modeling and Sentence Transformers. They showed that it is possible to find information on specific linguistic constructions in English in the output from BERT. This makes it likely that it is possible in other languages, such as Dutch, as well.

2.2 Word and Sentence embeddings

BERT [6] and models based on it, such as RoBERTa [14], BERTje [4], mBERT [6] and RobBERT [5] output word embeddings. Word embeddings are representations of words based on words that are found in the context of this word. Each word that is input into a BERT-based model gets a vector as output.

Sentence-BERT [19] is a model fine-tuned on BERT models. Instead of outputting a vector for each individual word in the input it outputs one vector for the entire input sentence. It does this by first combining the word embeddings, and is trained to give semantically relevant output, meaning that more semantically similar sentences will be closer to each other in the vector space. To combine the word embeddings, Sentence-BERT uses three strategies. The first is to use the output of the CLS-token. The CLS-token is added in BERT input at the beginning of each sentence. The output belonging to this token is representative of the entire sentence. The second strategy is to take the mean of all output vectors. The final strategy Sentence-BERT uses is to take the maximum of each output vector. Sentence embeddings can be used similarly to word embeddings, but represent the entire sentence instead of individual words. Sentence-BERT was originally trained on the BERT and RoBERTa models, but now Sentence-BERT models pretrained on many different BERT models are available.

2.3 Cluster Analysis Methods

This research will use cluster analysis methods focused on existing cluster and how well they cluster. Three of these methods are the Dunn Index, the Silhouette Coefficient and the Davies-Bouldin Index. Previous work has been done using these cluster analysis methods. The Dunn Index has been used by Oortwijn et al. [18] in their research on distributional semantic networks of philosophical terms, and Zhou and Bloem [22]. The Silhouette Coefficient has been used by Sai et al [16] and Layton et al [13]. The Davies-Bouldin Index has been used by Idrus et al [9].

3 METHODOLOGY

3.1 Data

The data used for this research will be extracted from the Lassy Groot corpus [11]. This is a corpus of Dutch sentences with automatically generated syntactic annotations. It is comprised of about 700 million words. The annotations of the sentences contain the syntactical relations between words in each sentence and the part-of-speech (PoS) tags of individual words.

The sentences containing the verbal 'te'-infinitival complement clause with and without the complementizer 'om', as well as the word 'niet', will be extracted from the XML files in the corpus using the XQuery language.

To make sure that the sentences without the complementizer 'om' will only contain sentences that would also be possible with the complementizer, we will be using GrETEL [17], a search engine for syntactically annotated corpora, to create a list of verbs that can take the 'te'-infinitival complement clause both with and without the complementizer 'om'. These verbs will then be used to select the sentences from the Lassy Groot corpus that include either of these clauses, or one of the verbs followed immediately by the word 'niet'.

This means that all the sentences that will be selected will contain one of the following forms:

- a verb from the list + *om* + optionally other words + *te* + infinitive verb
- a verb from the list + optionally other words + *te* + infinitive verb
- a verb from the list + *niet* + optionally other words

The selected sentences will be tokenized by the Dutch version of Python's NLTK package's WordTokenizer [1]. As will be described in the following section, the tokenized sentences will then be transformed into word embeddings by both the BERTje and the robBERT

models, after which these word embeddings will be turned into sentence embeddings.

3.2 Large Language Models (LLMs)

After these sentences have been extracted from the corpus and tokenized, the following LLMs, BERTje and robBERT, will be used to get word embeddings for each sentence. These word embeddings will be transformed into sentence embeddings.

3.2.1 BERTje. BERTje [4] is a Dutch version of the BERT model. It uses the same architecture and parameters as the original BERT and is trained on a diverse dataset of more than 2 billion words. The dataset contained, among others, news sources, wikipedia pages and books.

3.2.2 RobBERT. RobBERT [5] is also a Dutch version of BERT, but instead of it using the same architecture as the original BERT, it is based on RoBERTa [14]. RoBERta is an optimized version of BERT. RobBERT was trained on a significantly bigger corpus than BERTje with over 6 billion words.

3.2.3 Sentence Embeddings. The output from the BERTje and robBERT models are word embeddings. To be able to compare the different sentences with each other, we need to transform these into sentence embeddings. There are different ways to do this. The simplest way is to add the vectors from all the word embeddings. Other ways to combine the word embeddings are to take the mean of all the vectors, or to do max-pooling or min-pooling. BERT also provides a [CLS] token at the start of each sentence to represent the entire sentence. This could also be used as a sentence embedding. Another option is the Sentence BERT model [19]. The Sentence-BERT model is created specifically to fine-tune existing BERT models to output sentence embeddings instead of word embeddings. There are many pretrained versions of Sentence-BERT available, trained on different BERT models. Unfortunately, there is no sentence-BERT model available that is trained on the BERTje or RobBERT models. To be able to use a sentence-BERT model, we would have to train it ourselves on BERTje and RobBERT. All of these options will be taken into consideration and an informed decision will be made. Because the goal is to evaluate the output of the BERTje and RobBERT models directly, a simpler method is probably preferred.

3.3 Cluster Analysis

The sentence embeddings that will be the output of the Large Language Models will be high-dimensional vectors. The three linguistic constructions, the verbal 'te'-infinitival complement clause with complementizer 'om' and without complementizer 'om' and the verbs in combination with the word 'niet', will be taken as three separate clusters.

To be able to compare the influence of meaning and form of the linguistic constructions, we will divide the data into two groups. The first group will consist the sentences with both versions of the verbal 'te'-infinitival complement clause, with and without 'om'. Because the difference between these two groups is only in form and not in meaning, this group will be used to measure how well the clusters separate based on form.

The second group will also contain the sentences with the verbal 'te'-infinitival complement clause without 'om', but not the sentences with 'om'. This group will also contain the sentences with the verbs in combination with the word 'niet'. Since the addition of the word 'niet' changes the sentences to the opposite meaning, this group will be used to measure how well the clusters separate based on form.

These two groups containing two clusters each will be evaluated by calculating the following cluster evaluation methods: the Dunn Index, the Silhouette Coefficient and the Davies-Bouldin Index. The following sections will describe these cluster analysis methods.

3.3.1 Dunn Index. The first evaluation method that will be used is the Dunn Index [7]. The Dunn Index measures how coherent the clusters are, taking into account the inter-cluster distances and the maximum distance between members within a cluster. To calculate the distance between members cosine similarity will be used. A higher Dunn Index means that the clusters are better separated.

3.3.2 Silhouette Coefficient. The second evaluation method that will be used is the Silhouette Coefficient [20]. In this method, each cluster has its own silhouette. For each sentence embeddings vector, a silhouette value is calculated, comparing how the distance between this vector and the members within its cluster and the distance between this vector and members of other clusters. To calculate the distance between vectors, cosine similarity will be used. The value of a the Silhouette Coefficient varies between -1 and 1. The closer it is to 1 the more separated the different clusters are.

3.3.3 Davies-Bouldin Index. The final evaluation method that will be used is the Davies-Bouldin Index (DB Index) [3]. Like the Dunn Index and the Silhouette Coefficient, the Davies-Bouldin Index measures how well clusters coherence, using the separation between the clusters and the average distance of the members of a cluster to the center of that cluster. To calculate the distance between the sentence embedding vectors and the centroids cosine similarity will be used. A lower value means the clusters are better separated.

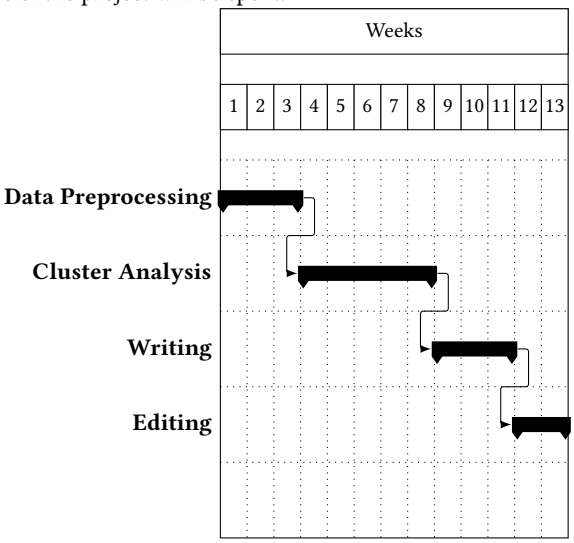
4 RISK ASSESSMENT

If there are technical issues with the Lassy Groot corpus due to its size, the Lassy Klein corpus [12] will be used. The Lassy Klein corpus is similar to the Lassy Groot corpus. It is much smaller, but it is still comprised of about 1 million words. Like the Lassy Groot corpus it contains Dutch sentences with syntactic annotations, such as syntactical relations and part-of-speech (PoS) tags. But in this corpus the annotations are not automatically generated, but manually verified. If there are technical issues with either of the Dutch BERT models, BERTje or RobBERT, they will be substituted for another Dutch Bert model, for instance mBERT [6]. mBERT is a multilingual version of BERT with 104 available languages, including Dutch.

5 PROJECT PLAN

The project will take place from March 31st 2025 until June 27th 2025. Before the official start of the project this thesis design and an Exploratory Data Analysis (EDA) will be concluded. The table 1

and the following Gantt Chart will show an overview of how the time of the project will be spent.



REFERENCES

[1] Steven Bird, Ewan Klein, and Edward Loper. 2009. *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

[2] Gosse Bouma. 2013. Om-omission in Dutch verbal complements. *Manuscript in preparation* (2013).

[3] David L Davies and Donald W Bouldin. 1979. A cluster separation measure. *IEEE transactions on pattern analysis and machine intelligence* 2 (1979), 224–227.

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

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

[6] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers).* 4171–4186.

[7] Joseph C Dunn. 1974. Well-separated clusters and optimal fuzzy partitions. *Journal of cybernetics* 4, 1 (1974), 95–104.

[8] Adele Goldberg and Laura Suttle. 2010. Construction grammar. *Wiley Interdisciplinary Reviews: Cognitive Science* 1, 4 (2010), 468–477.

[9] Ali Idrus. 2022. Distance analysis measuring for clustering using k-means and davies bouldin index algorithm. *TEM Journal* 11, 4 (2022), 1871–1876.

[10] Aniek IJbema. 2002. *Grammaticalization and infinitival complements in Dutch.* Netherlands Graduate School of Linguistics.

[11] Dutch Language Institute. 2023. Lassy Groot-corpus (Version 7.0) [Data set]. https://hdl.handle.net/10032/tm-a2-w8 Available at the Dutch Language Institute.

[12] Dutch Language Institute. 2023. Lassy Klein-corpus (Version 7.0) [Data set]. https://hdl.handle.net/10032/tm-a2-w9 Available at the Dutch Language Institute.

[13] Robert Layton, Paul Watters, and Richard Dazeley. 2013. Evaluating authorship distance methods using the positive Silhouette coefficient. *Natural Language Engineering* 19, 4 (2013), 517–535.

[14] 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. *arXiv preprint arXiv:1907.11692* (2019).

[15] Dang Minh, H Xiang Wang, Y Fen Li, and Tan N Nguyen. 2022. Explainable artificial intelligence: a comprehensive review. *Artificial Intelligence Review* (2022), 1–66.

[16] L Nitya Sai, M Sai Shreya, A Anjan Subudhi, B Jaya Lakshmi, and KB Madhuri. 2017. Optimal k-means clustering method using silhouette coefficient. (2017).

[17] Jan Odijk, Martijn van der Klis, and Sheean Spoel. 2017. Extensions to the GrETEL treebank query application. In *Proceedings of the 16th international workshop on treebanks and linguistic theories.* 46–55.

[18] Yvette Oortwijn, Jelke Bloem, Pia Sommerauer, Francois Meyer, Wei Zhou, and Antske Fokkens. 2021. Challenging distributional models with a conceptual network of philosophical terms. (2021).

[19] Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. *arXiv preprint arXiv:1908.10084* (2019).

[20] Peter J Rousseeuw. 1987. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics* 20 (1987), 53–65.

[21] Tim Veenboer and Jelke Bloem. 2023. Using collostructional analysis to evaluate BERT's representation of linguistic constructions. In *Findings of the Association for Computational Linguistics: ACL 2023.* 12937–12951.

[22] Wei Zhou and Jelke Bloem. 2021. Comparing contextual and static word embeddings with small data. In *Proceedings of the 17th Conference on Natural Language Processing (KONVENS 2021).* 253–259.

Week	Focus	Achievements
Week 1 (March 31 to April 4)	Data extraction	All data extraction will be done
Week 2 (April 7 to April 11)	Put data through LLMs	Sentence embeddings will be acquired
Week 3 (April 14 to April 20)	Writing Methodology section	Methodology section will be written
Week 4 (April 21 to April 25)	Cluster analysis preparation	
Week 5 (April 28 to May 2)	Dunn index	Dunn index scores will be calculated
Week 6 (May 5 to May 9)	Silhouette Coefficient and Davies-Bouldin Index	SC scores and DB Index scores will be calculated
Week 7 (May 12 to May 16)	Visualization of results	
Week 8 (May 19 to May 23)	Writing results section	Results section will be written
Week 9 (May 26 to May 30)	Writing Discussion section	
Week 10 (June 2 to June 6)	Writing Discussion section	Discussion section will be written
Week 11 (June 9 to June 13)	Writing Conclusion and Abstract sections	Draft Thesis will be written
Week 12 (June 16 to June 20)	Editing paper	
Week 13 (June 23 to June 27)	Finalizing paper	Final paper will be revised and edited

Table 1: Timeline of achievements within the project