Utrecht University
National Police Lab AI

Thesis Applied Data Science
INFOMTADS

An Implementation and Assessment of Semantic Search Few-Shot Classification

July 2, 2021

Universiteit Utrecht

Author:
S.A.C. Havermans (5974410)

External supervision:
D. Craandijk
J. Klepper

1st Supervisor:
prof. dr. A.P.J.M. Siebes

2nd Supervisor:
dr. M. van Ommen

Abstract

This thesis compares multiple methods of classification following cosine-similarity calculation from semantic search with Sentence-BERT (SBERT), as well as various class representations in few-shot classification with SBERT. The performance of SBERT is then compared to that of DistilBERT on various natural language processing (NLP) tasks (clickbait classification, sentiment analysis, spam detection and topic classification)[1] and datasets. This is done in an effort to determine for which tasks SBERT semantic search is an effective alternative to fine-tuning more traditional BERT models. The multilingual versions of both SBERT and DistilBERT are used for topic classification on a German dataset to assess performance of the multilingual version of SBERT. The best implementation of SBERT semantic search for few-shot classification uses a similarity-based classification as well as average embeddings for class representations. The results show that both SBERT and DistilBERT show signs of diminishing returns at around 25 samples per class when performing few-shot classification. Fine-tuning a DistilBERT model is equal to or outperforms SBERT semantic search on all assessed NLP tasks at a cost of slightly more instability.

[1]The results for sentiment analysis and spam detection are provided by K. Xie and O. Hsieh respectively, further reference on their contributions can be found in the Acknowledgements.

Contents

1 Introduction

In the current world, digitalisation is sweeping across organisations, making them more effective in the digital sphere. The Dutch National Police is no exception as an organisation that possesses copious amounts of data [1]. Extracting useful information from this data is crucial to the organisation's effectiveness. Currently, the Dutch National Police has two main approaches to extracting information from vast amounts of text. These include a syntactic approach as well as training a semantic natural language processing (NLP) model. The syntactic approach refers to analysis of natural language using the rules of formal grammar [2]. In the police context, this translates to using either a CTRL/CMD + F search for specific words or word combinations in a document or a standard database lookup [3]. Both the syntactic approach as well as training a model have their downsides and limitations. Using a syntactic approach in this manner limits the user to finding only specific words or word combinations, while not being able to identify semantics within the text. Although a trained semantic NLP model is able to identify semantics in text, it requires a large amount of labelled data and labelling data requires resources. Recent developments in the field of NLP might present a solution to this problem. Using a Transformer-based model called Sentence-BERT (SBERT) and semantic search it might be possible to directly use semantics in text for information extraction without the need to train an NLP model.

1.1 Literature Background

NLP is the field of research that focuses on using a computerised approach to analyse text and speech. Although the exact definition often changes due to it being a highly active area of research [4]. In 2017, Vaswani et al. [5] proposed the Transformer. In their aptly named paper 'Attention is All You Need', they proposed a model architecture based solely on attention mechanisms instead of using recurrence and convolutions. The essence of the Transformer lies within its use of positional encodings, attention and self-attention, which allow Transformers to consider the context of words, greatly improving performance on NLP tasks such as language translation [5]. From the Transformer architecture, a well-known family of NLP models was born: the Bidirectional Encoder Representations from Transformers (BERT). The BERT architecture is designed to pre-train deep bidirectional representations from unlabeled text on all layers, which then allows for fine-tuning only a single output layer to create highly effective models for a variety of NLP tasks [6].

The name BERT refers not only to a certain model architecture, but also to a pre-trained model itself. Using the BERT architecture, a whole host of NLP models have emerged, often using BERT in their name along with an indication as to what it was trained for. Pre-trained BERT models can differ from the original BERT in multiple ways. If the model is pre-trained on different data it is possible to produce BERT models in languages other than English or BERT models that are specialised for certain NLP tasks. Apart from the training data, they can also differ in their network structure itself [7]. One of the BERT models that fits into both categories

is SBERT, which uses a modification of the BERT network structure to derive semantically meaningful sentence embeddings from text [8].

Semantically meaningful in this context refers to semantically similar text embeddings being close to one another in vector space [8]. This allows for semantic search, where all entries within a corpus are embedded into a vector space. A query is then embedded into the same vector space, which will return the entries that are closest to it. The returned entries should then have a high semantic overlap with the original query. One important distinction to make in semantic search is the distinction between symmetric semantic search and asymmetric semantic search. Symmetric semantic search requires the entries and queries to have about the same length and content. An example of symmetric semantic search would be to find similar questions like "How should I fill up my pool?" and "How to put water into my swimming pool?". This is unlike asymmetric semantic search where a query is often short and consists of keywords to find a longer paragraph. Before any classification task is assessed using semantic search, it is important to specify which kind is being applied. Since entries and queries in symmetric semantic search are similar, it is possible to flip the two, this usually does not make sense for asymmetric semantic search. Using semantic search as an alternative to training semantic models used to not be a viable option. As performing semantic similarity search with classic BERT models (BERT, RoBERTa) creates an enormous computational overhead. Reimers and Gurevych [8] mention that finding the most similar pair in a collection of 10,000 sentences requires about 50 million inference computations which would take 65 hours with BERT or RoBERTa. SBERT however, is able to perform this task in roughly 5 seconds while keeping up with performance by deriving fixed-sized vectors from the input sequences [8]. This incredible leap in computational efficiency opens up semantic search as a possible alternative to training or fine-tuning an NLP model when trying to extract semantic information from textual data. If semantic search allows for the identification of semantically similar text when given examples of the target semantics, it could be possible to translate this semantic similarity into a classification of unlabeled text. Effectively using the semantics in text, which the syntactic approach cannot, while also not having to train an NLP model.

1.2 Problem Definition

The Dutch National Police requires a method to extract useful information from their data, that also requires as little resources as possible. First, In order to evaluate whether semantic search with SBERT is a useful method for information extraction in that context, an equal playing field is created between semantic search and a trained NLP classifier such as BERT. Semantic search, as the name suggests, is inherently based on producing similarity scores, while a trained BERT model is able to directly classify data. In order to create a fair comparison between the two, the similarity search of SBERT is turned into a classifier that is directly compared to the performance of a traditional BERT model. Due to limited computer resources, DistilBERT is used to represent the traditional BERT models such as BERT and RoBERTa. DistilBERT was created to allow the operation of a smaller general-

purpose language representation model. It is 40% smaller in size, 60% faster and retains 97% of the language understanding capabilities of the classic BERT model [9]. Making it the excellent candidate for the comparison between the unique BERT architecture SBERT and the more traditional BERT model structures.

The performance is measured on several NLP tasks relevant in the police context. These tasks include: clickbait classification, topic classification, spam detection and sentiment analysis. In line with the need for a low-resource method to extract information and thus a method requiring minimal labelled data, the performance on all these tasks are assessed under low training sample conditions, commonly referred to as few-shot learning or few-shot classification. Few-shot learning refers to a classifier quickly generalizing after seeing very few examples of each class [10]. If SBERT is able to outperform the traditional BERT models on classification tasks where training data is limited, this could prove to be a highly efficient method for the Dutch National Police to parse through textual data and extract relevant information, or to automatically classify large volumes of documents.

A dataset containing both clickbait and non-clickbait article titles is used as a benchmark during the creation of the level playing field between SBERT semantic search and training an NLP model (DistilBERT). The performance of the clickbait classification is also assessed during the comparison of both information extraction methods. Using clickbait to lure people into reading fake news publications or to make them click on suspicious links is common practice for bad actors. Their incentive might lie with the need to spread misinformation or to try and collect personal information from their victims [11]. The looming danger of a scam that sometimes lies within a clickbait title creates incentive for the Dutch National Police to be able to detect and possibly warn the Dutch people for these threats. Determining what the most efficient method is, referring to the amount of labeled data required, is therefore valuable information to have.

A second NLP task that is relevant in the police context is topic classification. Being able to determine what the content of a document entails without having to manually go over it can save valuable time out of someone's day. One example of an application of topic classification relates to stored documents on cold cases. These documents are often from a time before creating an online version was possible or common practice and therefore only exist offline. The Dutch National Police possesses a large amount of these physical documents from cold cases that are currently in the process of being scanned and added to the police database. Once it is possible to read the text of these documents directly with a computer, topic classification can help to label the documents into their relevant categories.

Finally, due to the Dutch National Police possessing mostly Dutch text data, the performance of multilingual models is assessed. Although using English data and models is easier during the formation of the equal playing field and the initial performance measurements, due to the much wider availability of data and models, it is important to create an indication of the performance on the more niche models

that are also able to process data in smaller languages.

Therefore, this thesis proposes an implementation of SBERT semantic search that allows for equal comparison between SBERT and a more traditional BERT classifier. This equal playing field is then used to answer the following three questions:

A How does SBERT semantic search perform on a clickbait classification task compared to a fine-tuned DistilBERT model under few-shot conditions?

B How does SBERT semantic search perform on a topic classification task compared to a fine-tuned DistilBERT model under few-shot conditions?

C How does SBERT semantic search perform on a topic classification task compared to a fine-tuned DistilBERT model in a multilingual setting under few-shot conditions?

2 Data

Three publicly accessible, labelled datasets are used in this thesis. These include the Clickbait dataset [12], the Web Of Science (WOS-11967) dataset [13] and a subset of the One Million Posts corpus [14].

2.1 Clickbait

The Clickbait dataset contains nearly 32 thousand article titles that each have a label of 0 or 1. A 0 denotes that an article title is non-clickbait, while a 1 labels the article title as clickbait. An example of a clickbait title would be: '10 Life-Changing Things To Try In November' or 'Niall Horan's New Glasses Are Causing Everyone To Lose Their Damn Minds'. Anyone who regularly surfs the web is likely to recognise these types of titles, where an amount of things that will shock you is named or where the illusion is created that something amazing must be presented in the content of the article. Unfortunately, this type of clickbait can be used to lure people into a scam as well, therefore it is important to be able to classify this type of language from a standard article title [11]. An example of a standard article title or a non-clickbait article title is: 'North and South Korea to hold second summit' or 'As Stimulus Piles Up Dollars, Their Value Falls'. These type of titles are more descriptive of what the content will directly be about and are less likely to engage readers under false pretenses. All examples mentioned here are directly sampled from the Clickbait dataset [12].

The article titles that make up the clickbait dataset are collected from a variety of news sources. The clickbait titles are collected from the news sources BuzzFeed, Upworthy, ViralNova, Thatscoop, Scoopwhoop and ViralStories. While the non-clickbait article titles have been collected from news sites such as WikiNews, New York Times, The Guardian, and The Hindu. The distribution of the labels in the dataset can be seen in Table 1, which shows that the data is nearly perfectly

balanced. The Clickbait dataset is freely available through Chakraborty's GitHub page [12].

Table 1: Label and number of samples in each category of the clickbait article title dataset [12].

Article titles	Label	Count
Non-Clickbait	0	16000
Clickbait	1	15986

2.2 Web Of Science

The data from the Web Of Science (WOS) dataset consists of published paper abstracts along with labels of the domain they were published in. In total there are 7 distinct domains that each have a numeric label assigned to them from 0-6 (Table 2). This data is used for topic classification, an NLP task that assigns a topic label to a text based on the content of that text. The amount of papers each domain contains is displayed in Table 2, which shows that it is a decently balanced dataset across the different classes, with Electrical Engineering and Civil Engineering being on the lower and higher ends of the spectrum respectively.

There are two critical differences between the clickbait data and the Web Of Science topic classification data. A paper abstract is much longer than an article title, which could possibly create problems when trying to use this data with DistilBERT as well as SBERT [8, 15, 16, 17]. Possible solutions to the length of the WOS data are using a different model, splitting the text into overlapping sequences or shortening the texts. Which solution is considered the most appropriate in this context is expanded upon in the Methods section. Examples of abstract samples from two domains (Computer Science and Mechanical Engineering) can be found in appendix A Data Examples for reference on what the data in the WOS dataset looks like. The second major difference has to do with the amount of classes. Clickbait classification is a binary classification task, while the WOS dataset contains 7 different classes. Binary classification only requires classification of a single class in the data. For example, if an article title is non-clickbait, this automatically means it is not clickbait and vice versa. Multiclass classification (a classification problem with 3 or more classes) requires a more specific indication as to what class a sample belongs to [18]. For example, if a sample from the WOS dataset is determined to not be Computer Science, it still leaves 6 other classes that the sample could belong to. A method that allows both binary as well as multiclass classification is therefore proposed in the Methods section.

The use of the Web Of Science data is "granted, free of charge, to any person obtaining a copy of this dataset" as per Kowsari et al. [13]. To make the data fit for use in this project, the X.txt and YL1.txt files from the WOS-11967 dataset were combined into the CSV format. Where X.txt contains the text and YL1.txt the labels referring to the domains the articles were published in.

Table 2: Label and number of samples in each category of the Web Of Science (WOS-11967) dataset [13].

WOS-11967	Label	Count
Computer Science	0	1499
Electrical Engineering	1	1132
Psychology	2	1959
Mechanical Engineering	3	1925
Civil Engineering	4	2107
Medical Science	5	1617
Biochemistry	6	1728

2.3 One Million Posts

Ideally, to answer research question C in a manner that is most relevant to the context of the Dutch National Police, a Dutch dataset would be used that was a direct translation from the WOS dataset. Unfortunately, there are no large publicly available clickbait or topic classification datasets available in Dutch. In order to still provide useful information, an approach using a German dataset is proposed. German, along with French, Italian, Spanish and Dutch was trained on the multilingual SBERT with very similar data. The training data consists of an English semantic textual similarity (STS) dataset that was translated using Google Translate and corrected by native speakers for each respective language. The benchmark sentence similarity tests performed by Reimers and Gurychev [19] show high similarity in performance between all of the abovementioned languages. Therefore, considering that the training method for these languages is the same, as well as the performance, showing that topic classification using SBERT semantic search works on German data could provide information on what the performance on Dutch would look like. Directly making it relevant for the Dutch text data from the Dutch National Police.

The One Million Posts dataset was originally intended to be a dataset concerning German user comments under articles from the Austrian news source DER STANDARD [14]. However, The metadata of the user posts also contains full articles with a maximum length of 750 words and topic labels representing what the articles are about. The dataset is licensed under the Creative Commons BY-NC-SA 4.0 License and is therefore freely available for use and augmentation as long as the appropriate credits are given [14].

A group named 'tblock' have subsequently published code under an MIT License on their GitHub page to subset the One Million Posts dataset and collect only the article contents and topic labels. This code produces the dataset referred to as the 10k German News Articles Dataset (10kGNAD). The 10kGNAD dataset contains a little over 10k news articles in German that belong to 9 different classes. Since 3 of the classes are much smaller than the average (Etat, Wissenschaft and Kultur), these and their articles have been removed from the data used in this thesis to produce the final subset of the One Million Posts corpus shown in Table 3. Examples from this subset can be found in Appendix A Data Examples.

Since this dataset is also used for topic classification, it presents the same challenges as mentioned for the WOS dataset. How the length of the samples and the multiclass classification is tackled, is expanded upon in the Methods section.

Table 3: Label and number of samples in the subset created from the 10kGNAD dataset that was originally split from the One Million Posts dataset [14].

One Million Posts	Label	Count
Web	0	1678
Panorama	1	1677
International	2	1511
Wirtschaft	3	1411
Sport	4	1201
Inland	5	1014

3 Methods

3.1 Few-Shot Definition

In order to make a fair comparison on few-shot classification performance across different models, datasets and NLP tasks, a singular definition is given to few-shot. Few-shot in this context translates to Equation 1, where X is the amount of samples per class, $Class_n$ represents all input data belonging to a specific label in the dataset and D_{train} is the resulting few-shot training data.

$$X(Class_1 + ... + Class_n) = D_{train} \tag{1}$$

This definition of few-shot allows both binary and multiclass classification problems, as D_{train} expands by X samples for every class that is represented within a dataset.

3.2 Sentence-BERT

SBERT, much like BERT, refers to both a model architecture as well as a collection of pre-trained models. Because of this, it is important to specify which pre-trained model was used specifically. On the repository created by Reimers et al. [8] the complete collection of pre-trained SBERT models is available. The tab for 'Semantic Search' specifies which pre-trained models are applicable for the symmetric semantic search tasks performed in this thesis such as clickbait and topic classification. These tasks are symmetric because, for example, a clickbait title is used as input to classify a remaining set of titles as either clickbait or non-clickbait, same length query and entry. For topic classification, the same is true, as a longer text is used as a query to identify similar sized texts as either belonging to the same topic class or not.

All references to SBERT in the clickbait classification task and the topic classification task with the WOS data refer to the pre-trained model 'paraphrase-distilroberta-base-v1'. This model was trained and tuned to embed sentences and paragraphs. The model is also used in many code examples and is relatively easy to use compared to some of the other pre-trained models [8].

Research question C requires a pre-trained model capable of evaluating German data. For the topic classification performed on the subset of the One Million Posts data, the model that is used is 'distiluse-base-multilingual-cased-v1'. This model is trained on similar data as is 'paraphrase-distilroberta-base-v1'. In fact, the German training data from the model is directly trained from an English semantic textual similarity dataset [20, 19]. Considering this and its relatively quick running time, which is relevant due to computational restraints during this thesis, it is used to represent the multilingual alternative to the previously used pre-trained SBERT model [8].

3.2.1 Cosine-Similarity Classification

As mentioned in the Introduction, semantic search natively produces similarity scores instead of assigning a label as a standard classifier would. In order to be able to assess the capability of semantic search to correctly classify entries into the desired class, an equal playing field has to be created. Meaning that SBERT semantic search has to be turned into a classifier that is directly comparable to the performance of a more traditional NLP classifier such as BERT. To achieve this, two methods of cosine-similarity classification are proposed and assessed: threshold-based classification and similarity-based classification. As the clickbait data contains short text sequences and is well balanced, this data will be used as a benchmark to determine which is the best out of the two. The near perfect balance in the clickbait dataset allows the accuracy to be used as a metric to score the success of each of these methods. Had the data not been as balanced as shown in Table 1, without accounting for the recall and precision the accuracy might have painted a skewed picture [21]. Compared to the WOS paper abstracts or the One Million Posts news articles, the titles are also very short, which requires less computational power and therefore allows for a quicker assessment of the cosine-similarity classification methods [12, 14, 13].

As the name indicates, threshold-based classification is based on the cosine-similarity between a labeled input sample and an unlabeled sample exceeding a set threshold. When this threshold is indeed exceeded, the unlabeled sample is labeled with the same label as the input example. Figure 1.a shows a schematic representation of what threshold-based classification would look like when a clickbait article title is used as an input example. When the cosine-similarity between the clickbait title and a sample exceeds the threshold, they are classified as clickbait. If they do not exceed it, the test samples are classified as non-clickbait. The few-shot setting does not allow for optimisation of the threshold, as that would require more labeled samples. Therefore, the threshold is determined after reviewing the cosine-similarities between the unlabeled samples and the input data. The threshold can

be set at a certain similarity level or in this case, knowing that the clickbait data is near perfectly balanced, is set at the mean similarity of all test samples. Having a threshold at the mean cosine-similarity when using a well-balanced binary dataset means that $\sim 50\%$ of the data will receive one label, which is what would line up with the actual labels of this dataset.

Similarity-based classification refers to calculating the similarities between a test sample and the representation of each class. The class representation that has the highest similarity to the test sample will then determine the label that is assigned to that test sample (Figure 1.b). Again, using the Clickbait data as example, the cosine-similarity is calculated between an unlabeled title and both a clickbait title and non-clickbait title. The two cosine-similarities are then compared and the highest similarity will determine which label is assigned to the unlabeled sample.

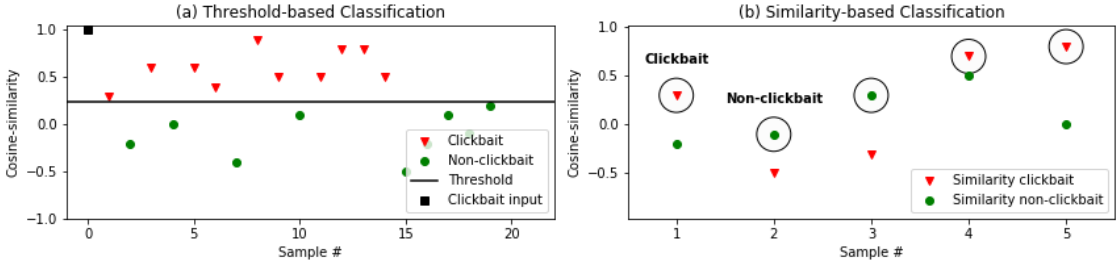

Figure 1: (a); A schematic representation of threshold-based classification. If the cosine-similarity between the input sample (in this example a clickbait article title) and a test sample is above the threshold (mean cosine-similarity) then the test sample will be classified as clickbait. If it is equal to or below the threshold it will be classified as non-clickbait. (b); A schematic representation of similarity-based classification. The cosine-similarity between a test sample and examples of all classes of the input dataset is calculated. The classification of the test sample is then determined through assignment of the label with which the highest similarity was observed, indicated by the black circles.

3.2.2 Few-Shot Class Representation

As defined by Equation 1, the size of the training data during few-shot learning is controlled by the size of X and the amount of classes. To evaluate the performance of both SBERT and DistilBERT under few-shot conditions, various values for X are assessed. For DistilBERT this means that X amount of samples of every class will be used to fine-tune the output layer of the model [9]. For SBERT however, there are multiple ways to create a representation of the few-shot data within the model.

One way to do this, is to create an average embedding for each class, referred to as a prototype representation and shown in Figure 3.a [22]. The idea is that this prototype representation has embeddings belonging to the same class clustered around it and would therefore be an effective way of representing the class as a whole [23]. As a result, calculating the cosine-similarity between any unlabeled sample and

the prototype representations of each class would lead to the highest similarity occurring between the test sample and the prototype representation of the class that it belongs to.

Another option of creating this prototype representation is by taking the median value in the embeddings. A prototype representation made up of median instead of average values might be better able to represent the class as a whole. The reason for this is mostly due to outliers [24]. Averages can more easily be influenced by outliers in the vector representations themselves, possibly pulling the prototype representation away from the center of the cluster that it is trying to represent [25]. Figure 2 shows a scenario where 3 embeddings are represented in a vector space with blue dots, the red triangle represents the mean prototype representation and the green square the median prototype representation. On the left side (a) of the figure, both representations are seemingly near the center of the cluster that they are supposed to represent. On the right side (b), the same points or 'embeddings' are shown, but with one additional outlier embedding. The triangle representing the mean prototype representation has moved away significantly from the cluster it is supposed to represent, while the median prototype representation has barely moved. Although this is a highly simplified presentation of the actual vector space that the embeddings are placed in, the concept of a median embedding is worth exploring. A schematic representation of the median prototype representation is shown in Figure 3.b.

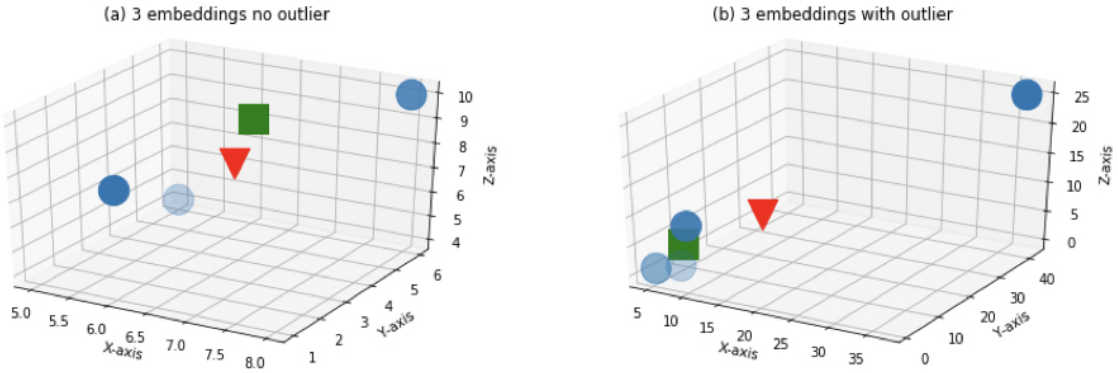

Figure 2: A schematic representation of a vector space with sequences embedded into them. The blue dots represent sentence embeddings as created by SBERT, while the red triangle represents a mean prototype representation of the embeddings and the green square a median prototype representation of the embeddings.

Finally, it is possible to not try and represent each class as a whole as input into SBERT, but to input every sample from the few-shot training data D_{train} into the model and compare them separately to the test samples. In this case, the classes would not be represented by prototype representations, instead they would be represented by each example within the class individually (Figure 3.c). Classification will then be determined by selecting the highest cosine-similarity from the similarities between unlabeled test sample and all embeddings within D_{train}, the class that the highest cosine-similarity belongs to will then determine the label that is assigned to

the test sample. This class representation will be referred to as complete representation and the classification using it as maximum similarity.

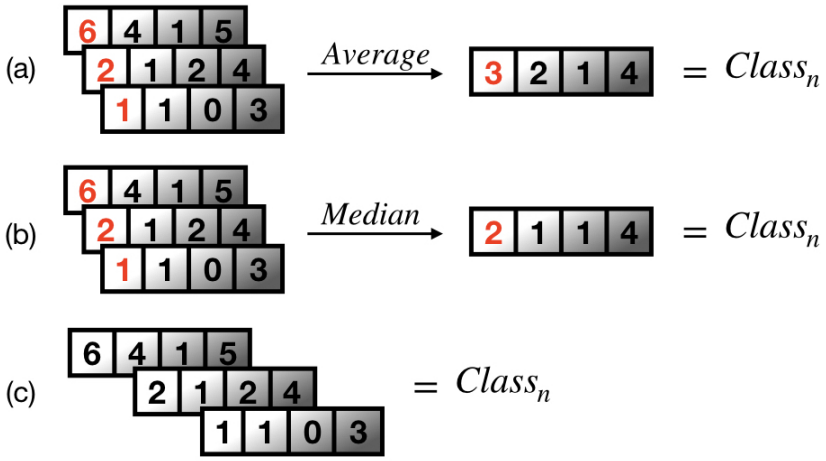

Figure 3: For every class in an input dataset a $Class_n$ representation is created through one of three methods. (a); A schematic representation of an average embedding prototype representation for $X = 3$. The red numbers indicate how the average is taken for the embeddings, as the average refers to a positional average within the embedding: $(6 + 2 + 1)/3 = 3$. (b); A schematic representation of a median embedding prototype representation for $X = 3$. The red numbers indicate how the median is taken for the embeddings, as the median refers to a positional median within the embedding: $1 < 2 < 6$, $Median = 2$. (c); A schematic representation of a complete representation for $X = 3$.

3.3 DistilBERT

For DistilBERT, the model 'distilbert-base-uncased' is used for both the clickbait classification task and the topic classification task with the WOS data. Like with SBERT, the One Million Posts data along with its topic classification task requires a version that can process German. Therefore, the 'distilbert-base-multilingual-uncased' is used [9].

3.3.1 Validation

DistilBERT models make use of three datasets during fine-tuning: D_{train}, D_{val} and D_{test}, which represent the training set, validation set and test set respectively. It is common practice to split the validation set from the test set, as it does not take away from the valuable training data [26]. In this scenario though, the objective is to identify how effective semantic search is compared to DistilBERT under few-shot conditions and it would therefore be an unequal comparison if DistilBERT was allowed to have much more labelled data to its disposal compared to SBERT through the validation set. Therefore, the validation set is split from the training data at a 3:1 split between D_{train} and D_{val} respectively. If $X < 4$, there will be no split and only a training set D_{train} and test set D_{test}.

Using this method of creating the validation set does come with exposure to a bias in the validation. At $X = 5$ there will only be a single validation sample for each class. If that one sample is a poor representation of the class as a whole, it might nudge the model in the wrong direction. One way to assess the amount of bias these extremely small validation sets bring could be through k-fold cross-validation. A method where the content of the validation set is differentiated over runs to assess prediction error [27]. For every run, a pool of labeled data dictated by the value for X is gathered randomly from the original datasets. From these, the training and validation sets are created while the remaining data is used as test data. Although this is not strictly k-fold cross-validation, it will create a similar effect where the content of the validation set is extremely likely to differ every run. The stability of the fine-tuned DistilBERT models is then assessed through the spread of results gathered from the various runs.

3.3.2 Parameters

The learning rate was set at 5e-5, as Sun et al. [28] show in their 2019 paper how more aggressive learning rates such as 5e-4 can cause BERT to suffer from catastrophic forgetting. A phenomenon where additional training of a pre-trained transfer learning model causes a sudden loss of pre-trained knowledge [29].

The maximum sequence length is set at 512 to accommodate as much information as possible without running out of memory on the GPU that the models are fine-tuned on. There are two other options to accommodate larger texts without running out of memory. It is possible to divide the texts into smaller, slightly overlapping sequences of about 200 tokens. The downside of this method is that it creates a large computational overhead [15]. Another option would be to use different transformers that natively accept larger sequences such as Longformer or Big Bird [16, 17]. Since the limit of sequence length is only relevant for the document classification task, the impact of setting the maximum at 512 is reviewed. From this, it is determined that less than 4% of the data in the WOS-11967 dataset would be affected by the limit. For the sake of continuity, DistilBERT will therefore be the model that is used together with a maximum sequence length of 512 [13]. Any sequences that exceed this limit are still used, but only starting from the beginning of the sequence until the maximum sequence length limit is hit. The same method is applied to the German One Million Posts/10kGNAD news article data, only the first 512 tokens are used as input for both SBERT and DistilBERT to allow for a fair comparison.

An early stopping rule is also implemented to try and prevent the model from overfitting on the low amount of training data that it is given each run. All DistilBERT models were trained for 10 epochs, with a patience of 3 on the validation loss. If the validation loss does not improve for the third epoch in a row, the fine-tuning is stopped. All other model parameters are left at their default settings.

3.4 Comparing Sentence-BERT & DistilBERT

Once the most effective and appropriate methods of cosine-similarity classification and few-shot class representations are determined, the stage is set for the equal playing field. It is now possible to start comparing SBERT semantic search and DistilBERT on few-shot classification performance (Figure 4). The stability of this performance is also evaluated by doing multiple runs for both NLP tasks and all datasets. For the binary clickbait classification task, 100 runs are performed for both SBERT and DistilBERT for all values of X out of the values: 1, 3, 5, 10, 25, 50, 100 and 250. For the document classification task with both the WOS and One Million Posts data, 30 runs are performed with both SBERT and DistilBERT for the same values of X. The accuracy from each of these runs is collected and output in tables and graphs containing the mean, min, max, $25^{th}\%$ percentiles and the standard deviation. The difference in amount of runs comes from the difference in computational requirements. The longer text samples that have to be processed in the topic classification data require more computational power and take significantly longer to run. 30 runs is therefore used as this should provide the results with enough reliability, while also keeping the total running time within reasonable bounds.

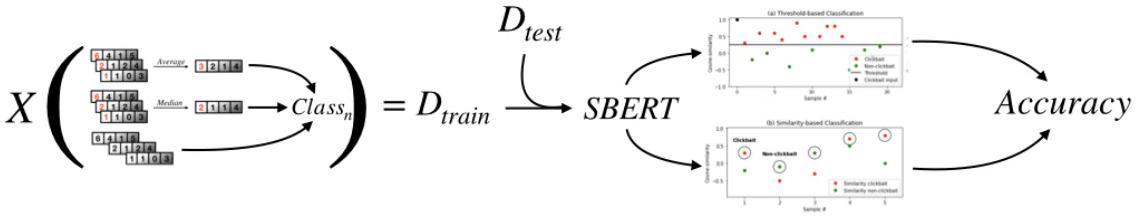

Figure 4: Schematic representation of the different methods that can provide a path from training data to accuracy using SBERT semantic search for few-shot classification. The left part of the image shows the various ways to create class representations (Figure 3) while the right side shows the two methods of cosine-similarity classification (Figure 1). An enlarged version of this image can be found in Appendix B Figures.

4 Results

4.1 Cosine-Similarity classification

Tables 4 and 5 show the results of both threshold-based and similarity-based classification for various few-shot settings; the results were gathered over 100 runs each for every setting and classification type. The highest average accuracy is achieved through threshold-based classification using clickbait inputs. Similarity-based classification is close to matching this performance. Using non-clickbait input examples results in worse than random guessing performance and gets increasingly worse as the amount of input examples is increased.

Table 4: The results for X=1 and X=5 few-shot settings for both threshold-based and similarity based classification. The threshold-based classification results contain 2 columns for every few-shot setting. This is due to the nature of threshold-based classification as well as the clickbait data itself. Clickbait classification is a binary classification task and due to working with the threshold it is only possible to input 1 class as an example at a time. For all clickbait (CB) columns, the input examples were clickbait, while non-CB columns had non-clickbait input examples. There are $2 \times X$ the amount of inputs, because similarity-based classification uses X amount of examples for every class and it is a binary classification task (Equation 1). This keeps the amount of input examples in both classification methods equal.

	Threshold		**Similarity**	**Threshold**		**Similarity**
	2 CB	2 non-CB	X=1	10 CB	10 non-CB	X=5
Mean	0.81	0.35	0.72	0.90	0.26	0.85
Std	0.047	0.102	0.063	0.019	0.071	0.030
Min	0.61	0.18	0.60	0.84	0.15	0.81
25%	0.78	0.27	0.66	0.88	0.21	0.83
50%	0.82	0.35	0.73	0.90	0.25	0.86
75%	0.84	0.42	0.77	0.91	0.30	0.88
Max	0.89	0.68	0.81	0.94	0.53	0.90

Table 5: The results for X=10 and X=50 few-shot settings for both threshold-based and similarity-based classification. The same considerations apply as referenced in the caption of Table 4.

	Threshold		**Similarity**	**Threshold**		**Similarity**
	20 CB	20 non-CB	X=10	100 CB	100 non-CB	X=50
Mean	0.91	0.23	0.90	0.93	0.19	0.94
Std	0.011	0.05	0.014	0.005	0.021	0.002
Min	0.87	0.15	0.88	0.91	0.15	0.94
25%	0.91	0.20	0.89	0.92	0.18	0.94
50%	0.91	0.22	0.90	0.93	0.19	0.94
75%	0.92	0.26	0.91	0.93	0.21	0.94
Max	0.93	0.40	0.93	0.94	0.25	0.94

4.2 Few-Shot Class Representation

The accuracy results for the various few-shot class representations are shown in Table 6. The performance of average embedding representation and maximum similarity far outrank the performance of the median embeddings. The performance between average embedding and maximum similarity is close although average embeddings do perform slightly better. Something that should also be considered in the applicability of these methods is the fact that maximum similarity takes considerably more time to run than average embeddings. Compared to average and median embeddings where a prototype representation is created, effectively creating a single embedding for each class, maximum similarity has X representations per class, as illustrated by Figure 3.c. Therefore, as you increase X, the running time increases by the same magnitude because the similarity is not just calculated for every $Class_n$ but for every $Class_n \times X$.

Table 6: Accuracy results for the clickbait classification task from the various few-shot class representations.

	Clickbait classification (5)			**Clickbait classification (10)**			**Clickbait classification (50)**		
	Average embedding	Median embedding	Maximum similarity	Average embedding	Median embedding	Maximum similarity	Average embedding	Median embedding	Maximum similarity
Mean	0.85	0.58	0.83	0.90	0.58	0.86	0.94	0.67	0.90
Std	0.030	0.064	0.032	0.014	0.058	0.022	0.002	0.040	0.011
Min	0.81	0.43	0.74	0.88	0.44	0.81	0.94	0.56	0.88
25%	0.83	0.53	0.82	0.89	0.58	0.85	0.94	0.64	0.90
50%	0.86	0.58	0.84	0.90	0.62	0.86	0.94	0.67	0.91
75%	0.88	0.62	0.86	0.91	0.65	0.88	0.94	0.70	0.91
Max	0.90	0.73	0.89	0.93	0.75	0.90	0.94	0.75	0.92

4.3 Few-Shot Classification Performance

The results of the cosine-similarity classification and the few-shot class representation give insight into what the best way of using SBERT in the few-shot setting is. From these results it is determined that for the cosine-similarity classification, similarity-based classification is best suited to create an equal playing field between SBERT and DistilBERT. Along with average embedding prototype representations this creates the classification pipeline for SBERT represented in Figure 5. That pipeline along with the few-shot definition and parameters for DistilBERT defined in the Methods section is used for the equal comparison between SBERT semantic search and a trained NLP classifier such as DistilBERT. Further reference on the decision on using these methods over others can be found in the Discussion and Conclusion.

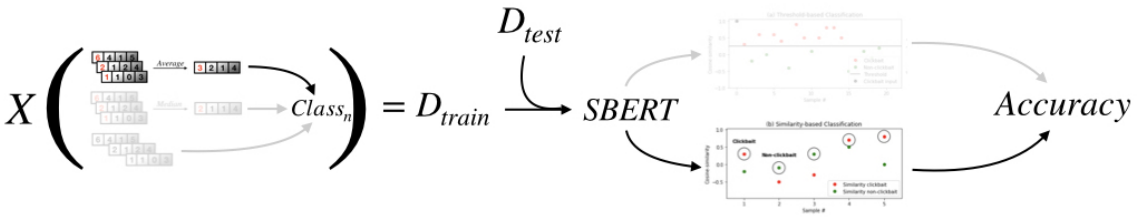

Figure 5: Schematic representation of the process from training data to accuracy using SBERT semantic search for few-shot classification. The left part of the image shows the various ways to create class representations (Figure 3) and the right side shows the two methods of cosine-similarity classification (Figure 1). The parts that have been determined less effective than possible alternatives have been greyed out. An enlarged version of this image can be found in Appendix B Figures.

The performances of both SBERT semantic search and DistilBert on varying few-shot settings can be seen in Tables 7, 8, 9 and 10. These tables contain results from four different NLP tasks, two of which have not been mentioned since the Introduction: sentiment analysis and spam detection [30, 31]. The accuracy results of sentiment analysis and spam detection are provided in full by K. Xie and O. Hsieh respectively, further reference can be found in the Acknowledgements.

Table 7: Few-shot ($X = 1$) SentenceBERT accuracy results compared to DistilBERT.

Few-shot (1)	**Topic classification**		**Sentiment analysis**		**Spam detection**		**Clickbait classification**	
	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT
Mean	0.41	0.53	0.50	0.52	0.69	0.69	0.72	0.73
Std	0.040	0.113	0.02	0.016	0.09	0.3	0.063	0.096
Min	0.34	0.25	0.47	0.50	0.54	0.18	0.60	0.51
25%	0.38	0.51	0.48	0.51	0.60	0.66	0.66	0.72
50%	0.40	0.58	0.50	0.52	0.70	0.72	0.73	0.74
75%	0.43	0.59	0.51	0.53	0.76	0.89	0.77	0.77
Max	0.47	0.62	0.52	0.56	0.83	0.98	0.81	0.86

Table 8: Few-shot ($X = 5$) SentenceBERT accuracy results compared to DistilBERT.

Few-shot (5)	**Topic classification**		**Sentiment analysis**		**Spam detection**		**Clickbait classification**	
	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT
Mean	0.59	0.71	0.52	0.53	0.8	0.937	0.85	0.86
Std	0.022	0.039	0.02	0.015	0.05	0.08	0.030	0.049
Min	0.56	0.66	0.48	0.49	0.66	0.79	0.81	0.78
25%	0.57	0.69	0.50	0.52	0.78	0.967	0.83	0.82
50%	0.59	0.69	0.51	0.52	0.81	0.971	0.86	0.87
75%	0.60	0.75	0.54	0.54	0.84	0.977	0.88	0.89
Max	0.62	0.76	0.55	0.55	0.89	0.98	0.90	0.94

Table 9: Few-shot ($X = 10$) SentenceBERT accuracy results compared to DistilBERT.

Few-shot (10)	**Topic classification**		**Sentiment analysis**		**Spam detection**		**Clickbait classification**	
	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT
Mean	0.65	0.83	0.52	0.55	0.87	0.943	0.90	0.89
Std	0.016	0.028	0.02	0.012	0.027	0.04	0.014	0.096
Min	0.63	0.77	0.50	0.53	0.82	0.877	0.88	0.68
25%	0.64	0.83	0.51	0.54	0.85	0.91	0.89	0.91
50%	0.65	0.83	0.51	0.55	0.87	0.95	0.90	0.93
75%	0.67	0.85	0.53	0.55	0.89	0.98	0.91	0.94
Max	0.68	0.87	0.57	0.57	0.92	0.988	0.93	0.97

Table 10: Few-shot ($X = 50$) SentenceBERT accuracy results compared to DistilBERT.

Few-shot (50)	**Topic classification**		**Sentiment analysis**		**Spam detection**		**Clickbait classification**	
	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT
Mean	0.70	0.91	0.54	0.58	0.94	0.98	0.94	0.90
Std	0.003	0.011	0.02	0.025	0.005	0.005	0.002	0.149
Min	0.69	0.89	0.52	0.55	0.92	0.976	0.94	0.49
25%	0.69	0.91	0.52	0.55	0.937	0.978	0.94	0.94
50%	0.70	0.92	0.54	0.58	0.939	0.981	0.94	0.95
75%	0.70	0.92	0.55	0.60	0.945	0.987	0.94	0.97
Max	0.70	0.93	0.57	0.61	0.95	0.988	0.94	0.97

Almost across the board, DistilBERT is less stable in performance than SBERT. With the standard deviations indicating that there is a larger spread of performance between runs than there is for SBERT. Stability has been an issue raised many times before with the fine-tuning of BERT models [26, 32].

Figure 6 shows that DistilBERT does outperform SBERT on every task except clickbait classification, where SBERT and DistilBERT are a close match for one another and both perform exceptionally well. What Figure 6 also shows is that having a binary classification task is not a guaranteed way to succeed with SBERT, as the sentiment analysis was also performed as a binary classification (positive/negative).

SBERT rather quickly reaches a point of diminishing returns on the spam detection, document- and clickbait classification tasks. The graphs show that between $X = 1$ and $X = 25$ there is a large improvement in accuracy, after which adding more samples per class leads to marginal or even no further improvement in performance. This effect is not seen in the sentiment analysis results where there is a very slight linear improvement as X is increased to 250.

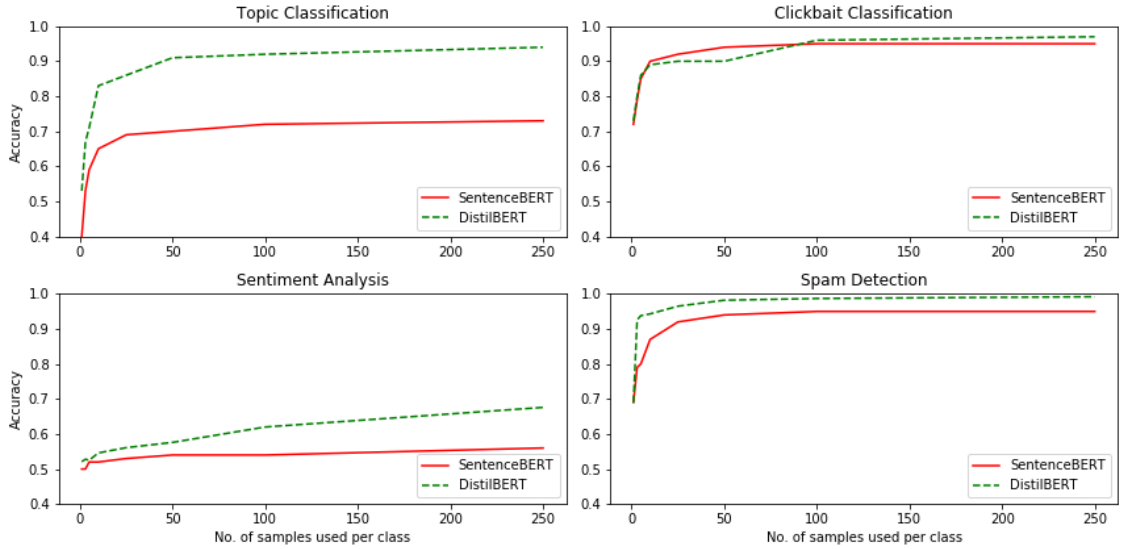

Figure 6: Few-shot classification performance on a variety of NLP tasks. The plot title indicates the task, the y-axis represents the accuracy on the respective test set and the x-axis represents the number of samples used per class, X.

Table 11 shows the accuracy results for the German One Million Posts news article data that was classified using the multilingual SBERT and multilingual DistilBERT. As with the English data DistilBERT seems both more unstable, but also able to achieve higher performances. regardless of the value for X.

The effect of diminishing returns with regard to performance and number of labeled samples can also be observed for the German news article data in Figure 7. Here too, the tipping point for those diminishing returns lies at roughly 25 samples per class ($X = 25$).

Table 11: Few-shot classification accuracy results with multilingual SBERT and multilingual DistilBERT on the German One Million Posts dataset.

Topic class.	$X = 1$		$X = 5$		$X = 10$		$X = 50$	
(German)	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT	SBERT	DistilBERT
Mean	0.49	0.51	0.65	0.70	0.74	0.75	0.80	0.84
Std	0.070	0.165	0.027	0.029	0.019	0.05	0.010	0.021
Min	0.38	0.13	0.61	0.64	0.70	0.61	0.78	0.80
25%	0.44	0.47	0.64	0.68	0.73	0.75	0.79	0.83
50%	0.49	0.57	0.65	0.70	0.74	0.77	0.80	0.84
75%	0.55	0.61	0.67	0.72	0.75	0.77	0.80	0.85
Max	0.58	0.68	0.70	0.73	0.76	0.78	0.82	0.86

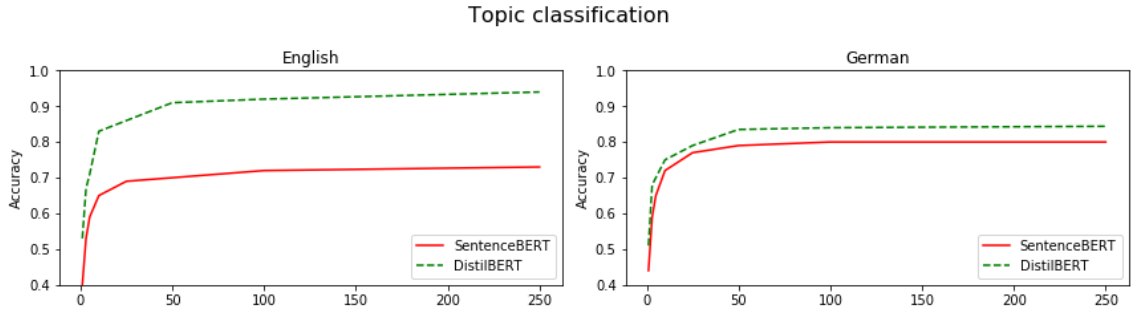

Figure 7: Few-shot classification performance on an English and German topic classification, with the language indicated by the plot title, the y-axis representing the accuracy on the respective test set and the x-axis representing the number of samples used per class, X. It is important to note that these accuracies were recorded on two different datasets (WOS for English and One Million Posts for German).

5 Discussion and Conclusion

A variety of conclusions can be drawn from the results obtained within this research project. These include conclusions regarding the method of cosine-similarity classification, discerning threshold-based from similarity-based classification. Conclusions regarding the few-shot class representation within SBERT to achieve the best results, while also considering running time and the computational overhead these representations create. As well as, perhaps most importantly, conclusions on the classification performance of SBERT semantic search under few-shot conditions in comparison to fine-tuning a more traditional BERT model.

5.1 Implementation for Classification

First, it is important to determine the best way to implement SBERT semantic search for classification and deduce why the methods in that implementation trump the others. Starting with the best method for cosine-similarity classification. Although the performance of threshold-based classification seemed to slightly outperform similarity-based classification. This section will outline why there is a strong argument for similarity-based classification being the superior method.

The first problem with threshold-based classification comes from its difficult application in multiclass classification problems, as with a single threshold only a binary outcome is produced. The similarity between an unlabeled sample and the input sample either exceeds the threshold (meaning the unlabeled sample belongs to the same class as the input) or does not exceed the threshold (meaning it does not belong to the same class as the input). For a binary classification these translate to sensible classes themselves (clickbait, non-clickbait). For any dataset that has more than 2 classes however, this is problematic. For example, classifying a WOS sample (Appendix A Data Examples) as either Mechanical Engineering or not Mechanical Engineering still leaves many questions as to what the content of the not Mechanical Engineering class is. Is the non-Mechanical Engineering sample from the domain of Computer Science? Biochemistry? etc. One could argue that there are ways to perform multiclass classification using thresholds. It is possible to set multiple thresholds on the cosine-similarity and create an ordinal classifier [33]. The problem then immediately becomes that this would only be possible with ordinal classes such as for example sentiment: Negative, neutral, positive. If, for example, the WOS data is used it would be impossible to create a sensible order out of the different scientific domains [13]. Additionally, the placement of the thresholds themselves would likely be difficult to determine, as few-shot learning inherently brings a lack of labeled data that is available for optimization of the thresholds [22].

The apparent sensitivity of threshold-based classification to the input class is what deals the second blow to this method of classification. Tables 4 and 5 show that the accuracy of clickbait classification drops sharply when a non-clickbait title example is used as input, as opposed to a clickbait title. This dependence on the input class is likely explained by the fact that clickbait article titles often have very distinguishable features: 'The Top 10 Most Exciting...', 'You Won't Believe What She Said When...' [12]. Whereas a class such as non-clickbait titles is much broader and is not as likely to have distinguishing features that can distinctly characterize it as such. Under few-shot conditions if non-clickbait titles are used as input these are likely to heavily bias the definition of a non-clickbait title. As an example, for the '2 non-CB' column in Table 4 if the two non-clickbait inputs are about dogs and flowers, any title that does not cover these two topics or something similar, are at risk of being classified as not non-clickbait in semantic search, which could explain the poor performance. Having all classes as input, as happens with similarity-based classification, could be an important rule of thumb to follow when using semantic search for classification with few labeled samples available. As in a genuine few-shot environment, it is not possible to check the accuracy as is done here, which would make it difficult to determine whether the right or most effective class was selected as input for threshold-based classification.

Similarity-based classification sports not only a high performance on the clickbait dataset, it also has inherent flexibility for different classification tasks. The amount of classes can theoretically be any amount and they can be both ordered or not. Therefore, similarity-based classification is the recommended form of classification when using SBERT semantic search for few-shot classification.

The few-shot class representations are the second critical aspect for the implementation of SBERT semantic search as a few-shot classifier. Luckily, the types of class representations lend itself to a more straightforward answer as to which type is the best to use. As there is one type that is both computationally the least intensive and has the best performance. Averaging the embeddings to create a prototype representation for each class far outperforms the equally intensive median embedding representation and performs slightly better than the computationally heavy complete representation. Therefore, using average embedding prototype representations is advised when using SBERT semantic search for few-shot classification.

The motivation for the assessment of median prototype representations next to the average/mean representation was mostly built on the potential negative effects of outliers within a class. In the few-shot setting these could easily skew the average embedding and could result in a poor class representation. The accuracy results of both prototype representations in Table 6 do not back this up. It might be the case that the classes do not contain outliers extreme enough to warrant the use of a median representation over the average. It is also possible that even if the clickbait dataset did contain outliers within its classes, that the input training data, which is selected at random from the whole dataset, never contained them. Further research into this idea could be done by introducing intentional outlier embeddings into a group of embeddings that is supposed to represent a single class. When a prototype representation is created through both median and mean representation the performance can be measured to see which prevails. One important thing to note here though, is that the results in Table 6 indicate that median representation is simply worse at approaching the center of the class cluster than the mean. With an accuracy almost 20% worse across various values for X, even if the median representation turns out to be more robust to outliers as represented in Figure 2, it still might not be worth it to use over the highly effective average embedding representation.

The complete representation along with its maximum similarity classification does show promising results, but is not able to match the performance boasted by the average embeddings. There are two reasons that might explain why. First, the idea behind a prototype representation is to put emphasis on the features that are characteristic to a class. To use the clickbait data as an example once more, clickbait article titles might be about a wide variety of topics: from celebrities, to vacations, to cute pets. What they all have in common are a certain vocabulary of a flashy nature in an attempt to get people to click [12]. By averaging the embeddings of a collection of clickbait titles, the one thing that is most likely to end up in the prototype representation is exactly those characteristic features. Therefore, when the similarity is calculated between that and an unlabeled sample and that unlabeled sample also contains these features, they are more likely to be close in vector space to the clickbait class representation and consequently be classified as clickbait. Complete representation not only lacks this distillation of clickbait features through averaging, the inherent non-clickbait language that any title contains, such

as the topic they are about, might also throw a wrench into things. A problem might arise when a clickbait input article title is compared to a non-clickbait title that both cover the same topic. With maximum similarity, a clickbait and non-clickbait title that both cover the same topic are likely to be close in vector space and as such, a wrong classification might be made. This is less likely to occur in an average embedding prototype representation because the specific topic of an article title is watered down through the averaging process over various titles.

5.2 Classification Performance

Now that an implementation of SBERT semantic search for few-shot classification has been proposed, it is important to look at what that means for the stated research questions. Starting with research question A that refers to the performance of SBERT semantic search on a clickbait classification task under few-shot conditions compared to fine-tuning DistilBERT. The top right plot of Figure 6 shows that DistilBERT outperforms SBERT semantic search in the long run, although their performances are closely matched. Something that is of note is the apparent dip of DistilBERT's accuracy at $X = 50$. This dip is a testament to a commonly seen problem with fine-tuning BERT models, their instability [26, 32]. DistilBERT seems to train very poorly for about 1 in 10 fine-tuning runs on the clickbait data. For the Dutch National Police, it might be worth it to use SBERT semantic search for clickbait detection as it will be easier to estimate how well it will perform. The same can be difficult to say about a fine-tuned DistilBERT model, because even though it is unlikely that the model is trained poorly and has poor performance, the risk of using a poorly trained model, might be worse than the slightly lower performance of semantic search few-shot classification. Combined with the results on spam detection, use of SBERT semantic search for few-shot classification is a viable option for detection of ill-intended text when few labeled samples are available.

The second research question, research question B, covers the performance on the NLP task topic classification. Here, SBERT struggles a lot more compared to the fine-tuned DistilBERT models, as indicated by the accuracies in Tables 7 - 10 as well as Figure 6. DistilBERT consistently outperforms SBERT semantic search on the WOS paper abstract data. On the German One Million Posts data the same is true, although by a smaller margin (Figure 7). The reason SBERT semantic search struggles here might be due to the difference in what SBERT was originally trained for and how it is applied in this context. Semantic search is trained to embed sentences or short paragraphs into vector space. These short paragraphs are defined as sequences with a maximum length of 128 tokens. The sequences that are embedded during the topic classification task regularly hit the set limit of 512 tokens. Semantic search with SBERT creates fixed-sized vectors of 768 dimensions regardless of the sequence length. This, among other factors, is what creates the computational efficiency that allows semantic search with SBERT when it was not computationally viable with more traditional BERT models [8]. It might be the case that squeezing information from these longer sequences into the fixed-sized vectors produces worse embedding representations compared to shorter sequences. Investigating the actual

embeddings that are produced when using sequences below the 128 limit and those produced from the WOS and One Million Posts data, something that might be interesting presents itself. The embeddings that are produced from sequences longer than 128 tokens contain values much closer to 0 than the embeddings produced at or below the 128 token limit (Figure 8). Further research would be required to find out whether this difference in embedding values is at the cause of the lack of performance of SBERT semantic search on topic classification. The multiclass nature of the datasets used for topic classification might also explain why the performance is lower, as classifying samples over 6 or 7 different classes is likely more difficult compared to binary classification, regardless of the sequence length.

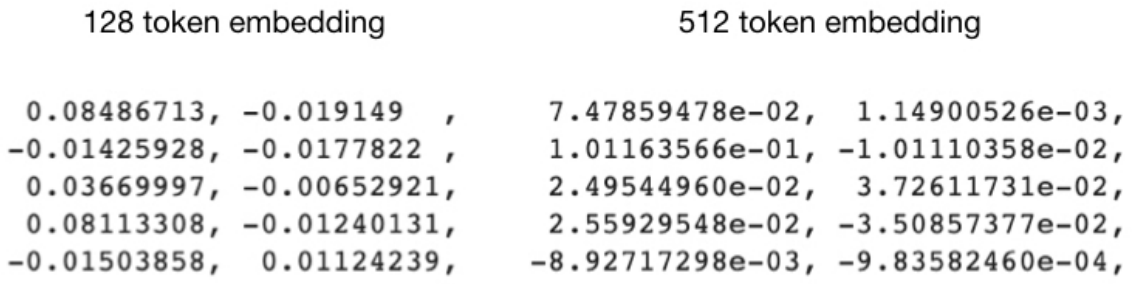

Figure 8: Two snippets of embeddings created by SBERT. The left embedding is from the first 128 tokens of the first sample in the WOS dataset, while the second embedding represents a 512 token sequence from that same sample. Note: embeddings are actually shape [768,] the partial view of these embeddings are for illustrative purposes only.

As it stands, SBERT semantic search is not likely to be the most useful method for topic classification for the Dutch National Police. Getting back to the cold case documents example given in the Introduction, SBERT semantic search in its current implementation lacks accuracy for reliable use of topic classification on these documents. The amount of wrongly classified documents would be high enough that employees of the Dutch National Police would still be required to manually review the outcome. Largely defeating the point of using semantic search in the first place. Training an NLP classifier such as BERT, possibly together with image classification could prove to be a strong and effective method for document classification of for example cold case documents [34].

Finally, research question C, which refers to the multilingual capabilities of SBERT compared to DistilBERT. This was assessed through topic classification using the German news articles from the One Million Posts dataset. The multilingual version of SBERT outperforms the topic classification performed using the English version. As the two datasets were different this does not mean that the multilingual version is better, the difference likely comes down to the content of the datasets. What it does mean however, is that the multilingual version of SBERT semantic search is capable of producing classification accuracies much higher than random guessing would achieve. As explained in the Data section, the German and Dutch language are pre-trained on the multilingual version of SBERT in the same manner as well as boasting highly similar performances on benchmark tests [19]. This, along

with the recorded German topic classification performance indicates that as of yet there is no reason to doubt that any conclusions drawn from the English SBERT experiments in this paper would not translate to a language such as Dutch. Further research is required to determine the gap, if any, in classification performance using the implementation proposed in this thesis between the multilingual model and the English SBERT. This would preferably be done by assessing the classification performance of multilingual SBERT and English SBERT on datasets translated from English to Dutch in order to make a more direct comparison.

5.3 Conclusion

The results of the semantic search few-shot classifications with SBERT are promising. Even though semantic search was not originally developed with classification in mind, the results show it can be very effective at doing so. Where semantic search is normally applied to semantic textual similarity tasks, scoring the similarity between sentences from 0-5, it is also highly effective at classifying spam or clickbait in a binary setting. Even on more complex classification tasks such as topic classification of the WOS data with 7 domain classes, the accuracy is much higher than random guessing would achieve. The aim of this project was to evaluate how few-shot semantic search with SBERT compares to fine-tuning a traditional BERT model. If it turned out that SBERT semantic search is highly effective at extracting relevant information from large volumes of text, this could increase the effectiveness of the Dutch National Police. They might be able to identify ill-intended advertisements on the web quicker or be able to classify large volumes of documents automatically, which would free valuable time of an officer or employee.

Reviewing all the results collected by K. Xie, O. Hsieh and those presented in this thesis, there are two important conclusions. SBERT semantic search is, in fact, able to produce high classification accuracy on a variety of NLP tasks. Namely, clickbait classification, spam detection and topic classification. If an employee of the Dutch National Police is inclined to find short sequences of distinguishable text from a collection of short sequences, SBERT semantic search is a viable option. Instead of syntactically searching for keywords, someone could provide a couple specific examples of the text they are looking for such as a clickbait title. Using SBERT semantic search they would be able to very quickly find similar text sequences with high accuracy and worry little about the stability of its performance. For longer texts or multiclass classification problems however, training or fine-tuning a semantic NLP model seems like the more viable option. All in all, SBERT semantic search as an information extractor under circumstances where few labeled samples are available is possible, but requires further research into implementation as well as performance for real perspective as the preferred method over training or fine-tuning a more traditional NLP model.

6 Acknowledgements

There are multiple people that have helped make this thesis possible. I would like to thank Dennis Craandijk and Judith Klepper from the National Police Lab AI and Arno Siebes from Utrecht University for their supervision and guidance in these unusual circumstances. Furthermore, I would like to thank Kelvin Xie and Oscar Hsieh who wrote their own theses on a related subject and gathered the data and results for the sentiment analysis and spam detection tasks. Without their help, I would not have been able to present these results in full or produce the efficiency in code that together we were able to. A collection of the annotated code used in this research project can be found at our thesis GitHub page.

7 References

[1] Politie.nl, "Het dataportaal van de politie." https://data.politie.nl//Politie/nl/. Accessed: 14-06-2021.

[2] P. Gorrell, *Syntax and parsing*, vol. 76. Cambridge University Press, 1995.

[3] A. F. Smeaton, "Using nlp or nlp resources for information retrieval tasks," in *Natural language information retrieval*, pp. 99–111, Springer, 1999.

[4] G. G. Chowdhury, "Natural language processing," *Annual review of information science and technology*, vol. 37, no. 1, pp. 51–89, 2003.

[5] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," *arXiv preprint arXiv:1706.03762*, 2017.

[6] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[7] N. Kitaev, S. Cao, and D. Klein, "Multilingual constituency parsing with self-attention and pre-training," *arXiv preprint arXiv:1812.11760*, 2018.

[8] N. Reimers and I. Gurevych, "Sentence-bert: Sentence embeddings using siamese bert-networks," *arXiv preprint arXiv:1908.10084*, 2019.

[9] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," *arXiv preprint arXiv:1910.01108*, 2019.

[10] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[11] E. Zeng, T. Kohno, and F. Roesner, "Bad news: Clickbait and deceptive ads on news and misinformation websites," in *Workshop on Technology and Consumer Protection (ConPro). IEEE, New York, NY*, 2020.

[12] A. Chakraborty, B. Paranjape, S. Kakarla, and N. Ganguly, "Stop clickbait: Detecting and preventing clickbaits in online news media," in *Advances in Social Networks Analysis and Mining (ASONAM), 2016 IEEE/ACM International Conference on*, pp. 9–16, IEEE, 2016.

[13] K. Kowsari, D. E. Brown, M. Heidarysafa, K. Jafari Meimandi, , M. S. Gerber, and L. E. Barnes, "Hdltex: Hierarchical deep learning for text classification," in *Machine Learning and Applications (ICMLA), 2017 16th IEEE International Conference on*, IEEE, 2017.

[14] D. Schabus, M. Skowron, and M. Trapp, "One million posts: A data set of german online discussions," in *Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, (Tokyo, Japan), pp. 1241–1244, Aug. 2017.

[15] R. Pappagari, P. Zelasko, J. Villalba, Y. Carmiel, and N. Dehak, "Hierarchical transformers for long document classification," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pp. 838–844, IEEE, 2019.

[16] I. Beltagy, M. E. Peters, and A. Cohan, "Longformer: The long-document transformer," *arXiv preprint arXiv:2004.05150*, 2020.

[17] M. Zaheer, G. Guruganesh, A. Dubey, J. Ainslie, C. Alberti, S. Ontanon, P. Pham, A. Ravula, Q. Wang, L. Yang, *et al.*, "Big bird: Transformers for longer sequences," *arXiv preprint arXiv:2007.14062*, 2020.

[18] M. Aly, "Survey on multiclass classification methods," *Neural Netw*, vol. 19, pp. 1–9, 2005.

[19] N. Reimers and I. Gurevych, "Making monolingual sentence embeddings multilingual using knowledge distillation," *arXiv preprint arXiv:2004.09813*, 2020.

[20] D. Cer, M. Diab, E. Agirre, I. Lopez-Gazpio, and L. Specia, "SemEval-2017 task 1: Semantic textual similarity multilingual and crosslingual focused evaluation," in *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, (Vancouver, Canada), pp. 1–14, Association for Computational Linguistics, Aug. 2017.

[21] Y. Liu, A. An, and X. Huang, "Boosting prediction accuracy on imbalanced datasets with svm ensembles," in *Pacific-Asia conference on knowledge discovery and data mining*, pp. 107–118, Springer, 2006.

[22] J. Snell, K. Swersky, and R. S. Zemel, "Prototypical networks for few-shot learning," *arXiv preprint arXiv:1703.05175*, 2017.

[23] A. Banerjee, S. Merugu, I. S. Dhillon, J. Ghosh, and J. Lafferty, "Clustering with bregman divergences.," *Journal of machine learning research*, vol. 6, no. 10, 2005.

[24] B. Justusson, "Median filtering: Statistical properties," in *Two-Dimensional Digital Signal Prcessing II*, pp. 161–196, Springer, 1981.

[25] P. T. Von Hippel, "Mean, median, and skew: Correcting a textbook rule," *Journal of statistics Education*, vol. 13, no. 2, 2005.

[26] T. Zhang, F. Wu, A. Katiyar, K. Q. Weinberger, and Y. Artzi, "Revisiting few-sample bert fine-tuning," *arXiv preprint arXiv:2006.05987*, 2020.

[27] T. Fushiki, "Estimation of prediction error by using k-fold cross-validation," *Statistics and Computing*, vol. 21, no. 2, pp. 137–146, 2011.

[28] C. Sun, X. Qiu, Y. Xu, and X. Huang, "How to fine-tune bert for text classification?," in *China National Conference on Chinese Computational Linguistics*, pp. 194–206, Springer, 2019.

[29] R. Kemker, M. McClure, A. Abitino, T. Hayes, and C. Kanan, "Measuring catastrophic forgetting in neural networks," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, 2018.

[30] R. Feldman, "Techniques and applications for sentiment analysis," *Communications of the ACM*, vol. 56, no. 4, pp. 82–89, 2013.

[31] N. Jindal and B. Liu, "Review spam detection," in *Proceedings of the 16th international conference on World Wide Web*, pp. 1189–1190, 2007.

[32] M. Mosbach, M. Andriushchenko, and D. Klakow, "On the stability of fine-tuning bert: Misconceptions, explanations, and strong baselines," *arXiv preprint arXiv:2006.04884*, 2020.

[33] E. Frank and M. Hall, "A simple approach to ordinal classification," in *European Conference on Machine Learning*, pp. 145–156, Springer, 2001.

[34] D. Song, A. Vold, K. Madan, and F. Schilder, "Multi-label legal document classification: A deep learning-based approach with label-attention and domain-specific pre-training," *Information Systems*, p. 101718, 2021.

A Data Examples

Web of Science (WOS) Computer Science (0) example:

The Langlands Programme, formulated by Robert Langlands in the 1960s and since much developed and refined, is a web of interrelated theory and conjectures concerning many objects in number theory, their interconnections, and connections to other fields. At the heart of the Langlands Programme is the concept of an L-function. The most famous L-function is the Riemann zeta function, and as well as being ubiquitous in number theory itself, L-functions have applications in mathematical physics and cryptography. Two of the seven Clay Mathematics Million Dollar Millennium Problems, the Riemann Hypothesis and the Birch and Swinnerton-Dyer Conjecture, deal with their properties. Many different mathematical objects are connected in various ways to L-functions, but the study of those objects is highly specialized, and most mathematicians have only a vague idea of the objects outside their specialty and how everything is related. Helping mathematicians to understand these connections was the motivation for the L-functions and Modular Forms Database (LMFDB) project. Its mission is to chart the landscape of L-functions and modular forms in a systematic, comprehensive, and concrete fashion. This involves developing their theory, creating and improving algorithms for computing and classifying them, and hence discovering new properties of these functions, and testing fundamental conjectures. In the lecture I gave a very brief introduction to L-functions for non-experts and explained and demonstrated how the large collection of data in the LMFDB is organized and displayed, showing the interrelations between linked objects, through our website www.lmfdb.org. I also showed how this has been created by a worldwide open-source collaboration, which we hope may become a model for others.

WOS Mechanical Engineering (3) example:

The development of printable biomaterial inks is critical to the application of 3D printing in biomedicine. To print high-resolution structures with fidelity to a computer aided design, materials used in 3D printing must be capable of being deposited on a surface and maintaining a printed structure. A dual-cross-linking hyaluronic acid system was studied here as a printable hydrogel ink, which encompassed both shear-thinning and self-healing behaviors via guest host bonding, as well as covalent cross-linking for stabilization using photopolymerization. When either guest host assembly or covalent cross-linking was used alone, long-term stable structures were not formed, because of network relaxation after printing or dispersion of the ink filaments prior to stabilization, respectively. The dual-cross-linking hydrogel filaments formed structures with greater than 16 layers that were stable over a month with no loss in mechanical properties and the printed filament size ranged from 100 to 500 mu m, depending on printing parameters (needle size, speed, and extrusion flux). Printed structures were further functionalized (i.e., RGD peptide) to support cell adhesion. This work highlights the importance of ink formulation and cross-linking on the printing of stable hydrogel structures.

One Million Posts/10kGNAD Web (0) example:

Eigener Browser Edge bei Suchen nach alternativen Browsern beworben. Es ist ein Running Gag in vielen Internetforen: Der Internet Explorer sei der beste Browser um Chrome oder Firefox herunterzuladen, so das Verdikt vieler Nutzer. Damit dies nicht so bleibt, versucht Microsoft nun die Macht der eigenen Suchmaschinen zu nutzen, wie Venturebeat aufgespürt hat. Wer im neuen Browser Edge nach Chrome oder Firefox sucht, bekommt als ersten Eintrag einen Werbeeintrag des Unternehmens für seinen eigenen Browser präsentiert. Mit diesem sollen Windows-10-User davon überzeugt werden, bei Microsoft Edge zu bleiben. Derzeit wird diese Nachricht offenbar nur für US-amerikanische Nutzer dargestellt. Und doch werfen einige Blogs mittlerweile Microsoft eine gewisse Scheinheiligkeit vor. Seit Jahren kritisiert Microsoft die vermeintliche Manipulation von Suchergebnissen durch Google. In der EU versucht das Unternehmen – mit Erfolg – auf Basis dieser Vorwürfe Untersuchungen gegen den Konkurrenten zu erreichen. Da mutet es zumindest seltsam an, dass man sich nun bei Bing genau der selben Praktiken bedient, die man Google bei seinem Shopping-Service unterstellt. Es ist nicht das erste Mal, dass Microsoft im Zusammenhang mit seinem neuen Browser Edge in die Kritik kommt. So hatte sich vor allem Mozilla lautstark darüber beschwert, dass mit Windows 10 der Wechsel auf einen anderen Browser deutlich erschwert werde.

One Million Posts/10kGNAD Inland (5) example:

Inflationswert von August 2014 bis Juli 2015, volle Abgeltung wie im Vorjahr. Wien – Die Pensionen werden im kommenden Jahr voraussichtlich um 1,2 Prozent angehoben. Dies ergibt sich aus der Inflation im relevanten Zeitraum und den gesetzlichen Vorgaben. Auch für heuer hatten die Pensionisten die volle Inflation abgegolten bekommen, nachdem sie zuvor zwei magere Jahre über sich ergehen lassen mussten. Der Anpassungsfaktor ergibt sich aus der durchschnittlichen Inflationsrate im Zeitraum von August 2014 bis Juli 2015. Dieser Wert liegt nach Berechnung der Statistik Austria bei 1,2 Prozent. Er muss nun noch von der Pensionskommission bestätigt werden, die im Herbst tagt. Sollte Sozialminister Rudolf Hundstorfer (SPÖ) davon abgehen wollen, wäre eine Gesetzesänderung nötig. Auch für heuer hatten die Pensionisten die volle Inflation von damals 1,7 Prozent abgegolten bekommen. In den beiden Jahren davor hatten die Pensionsbezieher Abstriche machen müssen. Mit dem Sparpaket 2012 wurde vereinbart, dass die Pensionen für 2013 um einen Prozentpunkt und für 2014 um 0,8 Prozentpunkte unter der Inflationsrate erhöht wurden. Für 2013 bedeutete dies eine Pensionserhöhung um 1,8 Prozent anstelle der vollen Inflationsabgeltung um 2,8 Prozent und für 2014 um 1,6 statt der Inflationsabgeltung um 2,4 Prozent. Die Mindestpensionen waren von diesen Kürzungen jedoch ausgenommen. Für 2015 und 2016 wurde in der Sparpakt-Vereinbarung von 2012 festgehalten, dass dann wieder alle Pensionisten die volle Teuerungsabgeltung bekommen.

B Figures

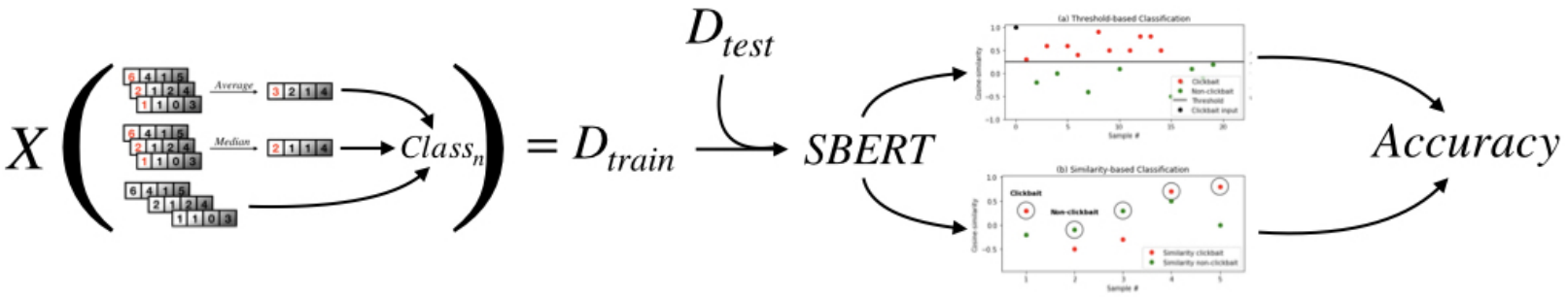

Figure 4: Schematic representation of the different methods that can go from training data to accuracy using SBERT semantic search for few-shot classification. The left part of the image shows the various ways to create class representations (Figure 3) and the right side shows the two methods of cosine-similarity classification (Figure 1). Repeated from page 13.

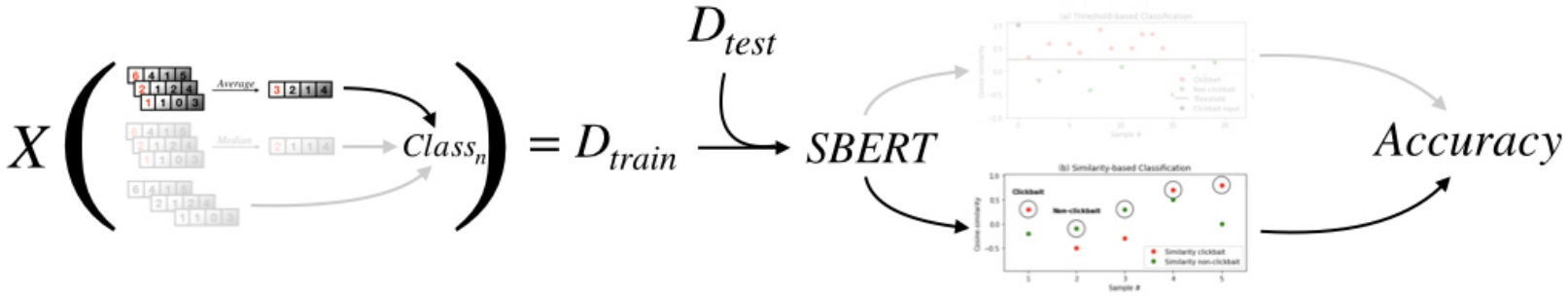

Figure 5: Schematic representation of the possible process from training data to accuracy using SBERT semantic search for few-shot classification. The left part of the image shows the various ways to create class representations (Figure 3) and the right side shows the two methods of cosine-similarity classification (Figure 1). The parts that have been determined less effective than possible alternatives have been greyed out. Repeated from page 15.