Universität Hamburg
DER FORSCHUNG | DER LEHRE | DER BILDUNG

Bachelor's Thesis

Evaluating Dimension Based Configurable Similarity Search

in the Language Technology (LT) group

by

Lars Lamberti

born on 13.02.1993

Matriculation number: 6799519

Field of study: Software-System-Entwicklung

submitted September 4, 2024

Supervisor: Hans Ole Hatzel & Tim Fischer

First Reviewer: Chris Biemann

Second Reviewer: Hans Ole Hatzel

Abstract

In this thesis we conduct a study on a configurable text similarity search, using a web interface along custom-trained SentenceTransformer text models. First, we provide a brief introduction to the topic, where we provide our motivation for this project along with formulating a problem statement. Can a similarity search improve the traditional approach for searches? Then we give an overview of related work in the fields of textual similarity search as well as the underlying technological structure in BERT and SBERT models. From there, we detail the configurable search methodology and our implementation in a web application. Participants in a study conducted as part of this thesis were asked to use the web application to perform a set of retrieval tasks. Finally, we evaluate the usefulness of similarity search with configurable dimensions and provide a conclusion for the test results as well as an outlook for future work in the field. While we were not able to prove a statistical significance to our results, we still made some interesting observations. Users were generally able to adapt to the different search approach quickly, shown in the retrieval times per task. The success rate of the similarity search was also above the traditional approach, however, this would need to be tested again on a bigger sample size to provide significant results. We also showed that there is a dimension focus depending on the given task, with geography, narrative and entities in a text being more valuable factors in news article similarity. Finally, we give an outlook to other possible use cases in and outside of the text domain, for example video game parts or music styles.

Contents

1 Introduction

In this chapter we first cover the motivation for our thesis and then move on to provide a clear problem statement for our experiments. Lastly we describe our three main research questions, before discussing some related work.

1.1 Motivation

With the amount of news generated today in various media, it is a common goal to identify how different perspectives will report on a singular event. A popular example of this is 'Ground News' [1], a website that gathers news articles from different outlets and categorises them based on their political tone. While a classical approach to this would be to study articles piece by piece, using regular free-text search engines, it requires some knowledge of available sources as well as the possible agendas that drive them. This task is very time-consuming and presumably inefficient. To provide a much more user-friendly solution to this, we want to introduce our configurable text similarity search. Our application will replace classic search engine functionality to instead find texts by certain factors or dimensions. This will give researchers a hub to aggregate and investigate different reports with less prior knowledge needed, while also being adjustable to the actual contents of the news in question. Also, this might uncover parallels that might otherwise be hidden in an article-by-article-based search. Furthermore, this can also be a useful comparison tool for text-based media other than just news articles.

1. https://ground.news/?fc=false

1.2 Problem Statement

Our main problems for this motivation are as follows: Given a user wants to research coverage on any topic, how can they find meaningful similar articles in a reasonable time? In a classical setup, they will likely refer to some key points to draw their conclusions. Thus they will need to identify certain factors of similarity to search for in different texts. This will lead to yet another problem in the field of interpretability, for example, if the user searches for all events related to a specific entity, it is not guaranteed they cover the same time period or geographical region, they might just be similar events that happened in completely different settings. Filtering out those that are not of interest at the moment will need more time and resources from the user, as well as some knowledge that the reports in question are actually not related. There is also very likely a barrier for articles in different languages since most researchers won't be able to actually cover every appearance from every country's correspondents reporting on it. These problems lead to our following main research questions.

1.3 Research Questions

1. **Does similarity search improve search efficiency**

To evaluate whether our approach is an actual improvement in time spent, as well as the correctness of results returned on any given submission we will closely analyze user performance. For this, we will measure the time it takes each participant to get to a set goal both for the classical text-based approach as well as our experimental setup. The number of legitimate results will also be taken as a measurement of the efficiency of the application, whereas legitimate results are made of both our target articles as well as similar ones solving a given task.

2. **Why do we need multiple dimensions of similarity?**

This is one of the core questions of this paper, which we will try to solve by setting up an experimental comparison in which a participant will use both a classical text-based search engine as well as our configurable similarity search to find related articles from a corpus of

texts. The assumption here is that several dimensions will be necessary to be able to filter out unwanted results, e.g. a dimension of time to narrow down events to a certain chronological period. This should improve the ability to aggregate articles over the classical approach by a large margin. This ties in directly to the next question:

3. **Which dimensions are needed and why should they be configurable?**

To answer this we will be monitoring the configurations of our users to see if and where there is a focus on certain factors, an entity based factor will presumably see more usage as opposed to a dimension for style of writing. This might also provide insight on dimensions we have not yet covered that might further improve usability of the application. The configurability will play a big role in getting streamlined results across all participants. It will also improve the ability to search for reports from different languages.

2 Background

In this section, we cover some of the concepts used throughout this thesis. We will introduce transformer architecture and explain the different encoder types, which paves the way for more specific information in the related works section.

2.1 Transformer Architecture

Transformers were initially proposed by Vaswani et al. (2023) in an attempt to completely replace previous state-of-the-art models based on recurrent and convolutional neural networks (RNNs & CNNs), with a new approach based only on attention mechanisms. While the top-performing models of those established architectures also employed attention mechanisms, Transformers rely solely on multi-head self-attention without the need for any recurrence or convolutions. Attention in this context is the idea of reducing computational cost by weighing and prioritizing ('paying attention to') relevant information (Soydaner, 2022). Transformer models still follow an encoder-decoder structure utilizing self-attention on each of the fully connected layers. The structure maps an input sequence of symbol representations x to a sequence of continuous representations z and generates an output sequence y given z, one element at a time (Vaswani et al., 2023). The attention functions in the model are then used so that the encoder can attend to all positions of the previous layer of the encoder and similarly the decoder can attend to all positions in the decoder and the auto-regressive property of the model is preserved. Vaswani et al. showed that Transformers are a major improvement to previous models especially in the field of text translation, but also applied to other tasks (English constituency parsing) even without

specific tuning. We will cover the models specific to our use case (BERT & SBERT[1]) in more detail in the next chapter.

2.2 Bi-Encoders vs. Cross-Encoders

For the encoder, there are two relevant structures in the Sentence Transformers (or SBERT) library we want to contrast here. First off we have Bi-Encoders, which are given sentences and from those they produce sentence embeddings, which can then be compared with cosine similarity. Cross-Encoders will receive both sentences simultaneously and return a similarity value between 0 and 1 for the pair[2](Reimers and Gurevych, 2019). Cross-Encoders are not suited for Semantic Search, as the time requirement does not scale well to large datasets and they do not produce embeddings, so computing actual similarity scores takes a long time. Bi-Encoders however, while achieving slightly worse performance, can compute embeddings and so are much faster. Other tasks can then be performed with that data. It is possible to combine the two encoders however, using the Bi-Encoder to produce embeddings and then use a Cross-Encoder to score and re-rank those results.

2.3 Vector Similarity

When working with Semantic Similarity, we need means to actually compare the similarity of one input to another. For this, the concept of vector similarity is used, since the embeddings we compute from our input will be represented by vectors. There are several measurements for this, like Manhatten or Euclidean distance (Reimers and Gurevych, 2019), however typically the cosine-similarity is used. This calculates the cosine of the angle between two vectors and produces an output from -1 to 1, where 1 denotes an identical vector, -1 for non-similar vectors and 0 for orthogonal vectors (Singhal, 2001). This is also the calculation we use to determine similarity between news articles.

1. https://www.sbert.net/index.html
2. https://www.sbert.net/examples/applications/cross-encoder/README.html

3 Related work

BERT for Language Understanding (Devlin et al., 2019)

In this paper by Devlin et al. (2019) some of the fundamental groundwork for our thesis is covered. It introduces the BERT language model, which is an acronym for Bidirectional Encoder Representations from Transformers and is part of the models we will be using in our experiment (Language-agnostic BERT sentence embeddings or LaBSE (Feng et al., 2022)). BERT features a fine-tuning strategy to apply pre-trained representations to downstream tasks, as opposed to a feature-based approach. Common fine-tuning approaches for pre-training feature unidirectional language models to learn representations. BERT aims to provide major improvements in this area by switching to a "masked language model" training function, which allows for deep bidirectional training, as well as a "next sentence prediction" for increased understanding of sentence relationships. To this end, some of the input tokens are masked and then predicted, otherwise a bidirectional language model could trivially predict each token. BERT was able to outperform many task-specific architectures and achieve new top results for eleven NLP tasks, including a 80.5% GLUE score, 92.3 F1 on SQuAD v1.1 and 83.1 F1 on SQuAD v2.0. (Devlin et al., 2019).

Computing Sentence Embeddings Using Siamese BERT Networks (Reimers and Gurevych, 2019)

The work by Reimers and Gurevych (2019) further expands on this area by introducing SBERT, a modification of BERT that introduces Siamese network structure to achieve semantically meaningful sentence embeddings. This makes it much more suitable for tasks like semantic similarity search or clustering, as it greatly reduces the computational requirements and the time needed

to a fraction of BERT computations. SBERT achieves this by generating fixed-size sentence embeddings through pooling operations on BERT output, which can then efficiently be evaluated using cosine similarity. Three strategies are considered and evaluated over several semantic textual similarity (STS) tasks. Additionally, three different structures and objective functions are considered. The Classification Objective Function concatenates sentence embeddings with their element-wise difference and multiplies this with trainable weights, then cross-entropy loss is optimized. The Regression Objective Function computes cosine similarity between two sentence embeddings and uses mean squared error loss. Finally, the Triplet Objective Function tunes the network by computing smaller distances between a given anchor sentence and a positive sentence than the distance between an anchor and a negative sentence. Similarities are mostly compared by their cosine-similarity to keep it scalable on big datasets. The classification objective function instead multiplies the sentence embeddings and their element-wise difference with the training weights and optimizes cross-entropy loss. SBERT is able to achieve higher scores than other state-of-the-art sentence embedding methods, improving on most unsupervised tasks (STS12-STS16, STSb) while only falling behind the Universal Sentence Encoder (Cer et al., 2018) on the SICK-R dataset. For the supervised STSb test, SBERT-base was only able to achieve minor improvements over BERT when trained on the dataset and falls behind BERT when further trained on the NLI dataset. It achieved mixed scores on other downstream tasks like the Argument Facet Similarity (AFS) where it fell behind BERT, but improved scores on the task by Ein Dor et al. (2018) about Wikipedia sections. These studies show that standard BERT is unsuited for common STS tasks, with the exception of the SentEval task, as the purpose of SBERT is not to be used for transfer learning. Here the average BERT embeddings actually provide decent results, since certain dimensions can have different weights in regards to the result. For large sets of data however, BERT is unfeasible since the computational requirements are significantly higher than SBERTs, reducing time for an example dataset of 10.000 sentences with hierarchical clustering from 65 hours to 5 seconds (Reimers and Gurevych, 2019)

Multilingual News Article Similarity (Chen et al., 2022)

This paper by Chen et al. provides us with the SemEval-2022 Task 8, the source of the dataset which we will be using for our own experiments. They state the task to assess similarity of news articles across several dimensions in a multilingual corpus. The underlying problem is based on those dimensions and how they interact, e.g. two news articles describing similar events in different time periods, think news coverage of football events for example. For humans, this information is mainly transferred through context, which is not available to language models in that form. The dataset contains about 60M articles in 10 languages pre-filtering, with the processed dataset landing around 23,5M. Articles were filtered by missing metadata, length of text and relevancy, with social media explicitly excluded as well. Those articles were then sampled and matched with another article to form meaningful pairs. Then, guidelines were formed by which human annotators would classify each article pair on a four-point Likert scale, in regards to the seven dimensions:

- Geography: How similar is the geographic focus (places, cities, countries, etc.) of the two articles?
- Entities: How similar are the named entities (e.g. people, companies, organizations, products, named living beings), excluding previously considered locations appearing in the articles?
- Time: Are the two articles relevant to similar time periods or describing similar time periods?
- Narrative: How similar are the narrative schemas presented in the two articles?
- Overall: Overall, are the two articles covering the same substantive news story? (excluding style, framing and tone)
- Style: Do the articles have similar writing styles?
- Tone: Do the articles have similar tones?

geographic focus, named entities, time periods, narrative schema, overall substance, writing style and tone.

With the labeled dataset ready, the task was advertised alongside other SemEval2022 tasks. Chen et al. provided baseline models using SVC (Support Vector Classification) with linear kernel, logistic regression, random forest and XGBoost (Chen and Guestrin, 2016). Those would then evalute three sets of features, mainly focused on Jaccard similarity of named entities and full text. The results showed, that most entries to the task performed better than those baseline models. Participants that worked on fine-tuning embeddings were generally more favorable over those that did work on the features. However, there were mixed reports on the superiority of the transformer architecture, with some proving that bi-encoders would outperform cross-encoders and others proving the contrary. The multilinguality of the dataset would be best solved by employing combinations of mulitingual embeddings and translation of articles. At the end of the task there was still no clear consensus over which approach works best, as there has been conflicting evidence on the top subjects in almost each field, architecture, embeddings models or preprocessing of data.

Dimensions of Similarity: Towards Interpretable Dimension-Based Text Similarity - European Conference on Artificial Intelligence 2023 (Hatzel et al., 2023)

The work by Hatzel et al. provides the base for this thesis. It serves a first approach to what a configurable dimension-based similarity search could look like. The focus here lies on the different dimensions as designed by the SemEval 2022 Task 8 (Chen et al., 2022) and to extract interpretable results from using semantic similarity search. The hypothesis here is that existing models focus on named entities for stability, however it is suggested that in certain cases this is counterproductive. Instead, models for each of the seven dimensions are created and evaluated, with the goal of a user-configurable similarity search. The models feature a bi-encoder structure, as cross-encoders are not suitable for large datasets simply due to time requirements. Furthermore, the base model used to train each dimension on was chosen from a selection of the entries to SemEval 2022 Task 8, with LaBSE being the one that has the best balance between complexity and overall performance on multilingual documents. This was

then compared to a baseline entity focused model which did outperform an untuned LaBSE model, which proves the theory of an entity focus to some extent. During the fine-tuning, it was evaluated that the multi-model (MM) approach that trains each dimension in a separate model greatly improved the pre-trained model and even outperformed another multi-task-learning (MTL) approach, where all dimensions get trained into one model. This is in part due to the fact that some of the dimensions have a strong inherent correlation. Comparing human and machine judgement on inter-dimension correlation demonstrates this clearly. Machines have trouble distinguishing certain dimensions, or – in the case of the MTL model – between the majority of dimensions, while humans only run into difficulties with dimensions that do actually correlate. The results of the downstream experiments conducted by Hatzel et al., in both document retrieval and classification tasks, show promising results towards a user-configurable similarity search. While some dimensions worked well on tasks outside of the SemEval 2022 Task 8 dataset, it is still not applicable to completely unrelated domains. The conclusion leads directly into our thesis of creating a configurable similarity search system to perform experiments with human users on real-world use cases.

4 Experiments

In this chapter, we will first explain in detail how the dataset was constructed, which fields we defined for our articles and where we sourced them from. The second part is going to cover the architecture we built for both front- & back-end for both of our search interfaces. Afterwards, we describe the idea and computations behind our study before finally illustrating an example for possible use cases.

4.1 Dataset

The dataset that builds the base for our research is the SemEval 2022 Task 8 dataset which consists of news articles in 18 different languages taken from the web, annotated accordingly with a central line of question for each semantic aspect or dimension of the text, e.g. 'Do the articles describe similar places in space?' for geography and so on. The dimensions available focus on time, geography, narrative of the text, style of writing, tone, (named) entities, as well as an overall dimension. It is separated into a smaller training subset (with less language variability) and a bigger evaluation set. We will be using the full evaluation dataset for as much variance as possible. Furthermore, since the dataset is from 2022 and not fully compatible with some of our study cases regarding certain news events, we added onto the dataset another ~6.000 articles we scraped from German news outlets for a specific time frame. For this we utilized the open-source crawler "News Please" by Hamborg et al. (2017). We ran its script dedicated to gather articles from the Common Crawl news archive, limiting the set to only a few major German publishers, ranging in dates from 2020-2023. This should return more search results that might at first glance be relevant, making the retrieval task more challenging, especially in

the classic text-based search setting. So in total, the set is made up of 16.666 news articles, roughly 10.000 of which stem from 18 languages, with our roughly 6000 German articles added. Each article will be processed to only contain the necessary fields:

- 'body': the main text (including introductory notes) of the article
- 'title': the main headline of the article
- 'publish_date': the date when the article was published, as precise as possible
- 'article_id': a generated string of numbers of length 10 for identification

4.2 Architecture

To create a configurable similarity search, we are using a pre-trained model for multi-lingual data, which we then further fine-tuned on our seven semantic dimensions of similarity to achieve a meaningful representation. The base model we chose is LaBSE (Feng et al., 2022) which produces language agnostic, bidirectional sentence embeddings, which is a good baseline (see Table 1 in Hatzel et al. (2023)) for our fine-tuning on multi-lingual texts. We use 7 variants of the model each fine-tuned by Hatzel et al. (2023) on one of the dimensions of similarity as defined by Chen et al. (2022). Provided with these models, we can now begin to build an experiment setup.

For this purpose, we create a simple web application (see Figure 4.1). We chose vue.js [1] as our framework because of its effective design in terms of routing and front-end development. We go with a single-file component architecture, building each part of the application independently and then joining them together in our view templates. The application is built with 4 main components: The dimension/querying bar on the left side, which features adjustable range sliders from -1 to 1, the ranges on our cosine-similarity computations for each dimension. Alongside those are tooltips explaining each dimension further. Once a configuration is made,

1. https://vuejs.org/

the user can hit the compare button to show results sorted by their similarity to the base article, or hit reset to set all sliders back to 0. Furthermore there is another more detailed explanation of the feature. The middle component is our results list. There we present the most similar articles in reference to the configuration given by the user. Each entry in this list has some of the data shown (title + body preview) and a button to show the article in full as well as giving the user the option to submit any article for a given study question. This will be presented in a modal window, see Figure 4.2. Additionally, we show the computed similarity score for a given article in the list. For this we compute the following:

$$similarity = \begin{pmatrix} t_1 \\ t_2 \\ \dots \\ t_n \end{pmatrix} x \begin{pmatrix} w_1 \\ w_2 \\ \dots \\ w_7 \end{pmatrix} = \begin{pmatrix} r_1 \\ r_2 \\ \dots \\ r_n \end{pmatrix}$$

With the first vector representing the calculated cosine similarity between the base article and all other (pre-computed) embeddings, the weights tensor and a result tensor. All the result entries have a summarized score based on the weights vector. This is additive, meaning the score will go higher if the number of dimensions that are used increases.

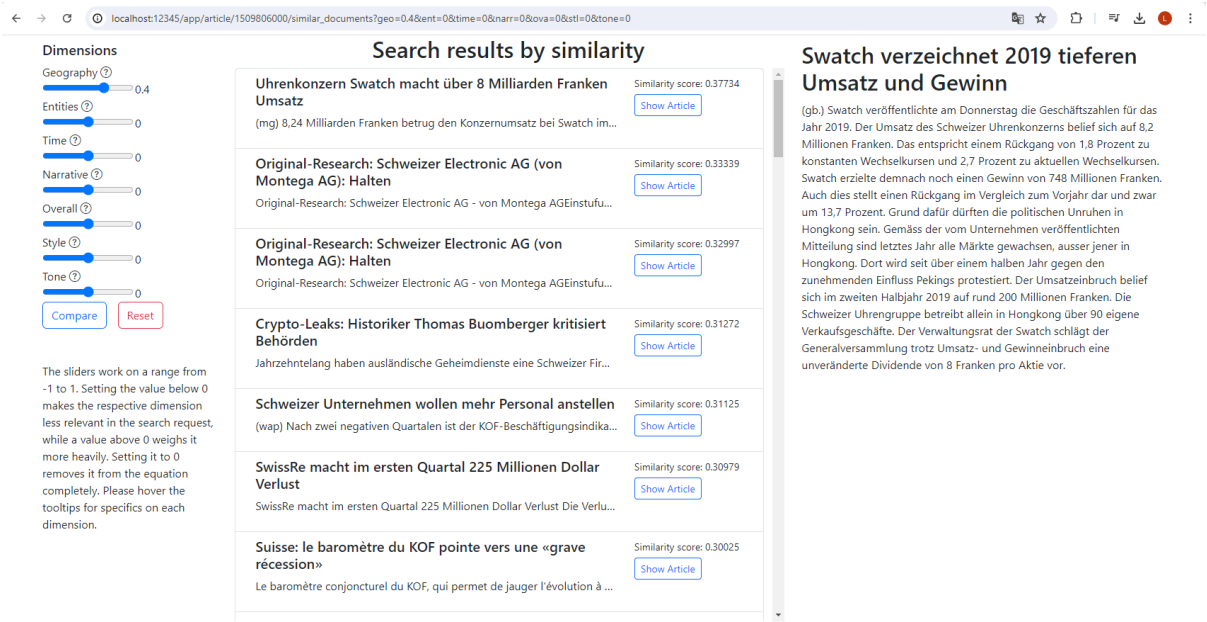


Figure 4.1: Prototype of the web application for our experiment

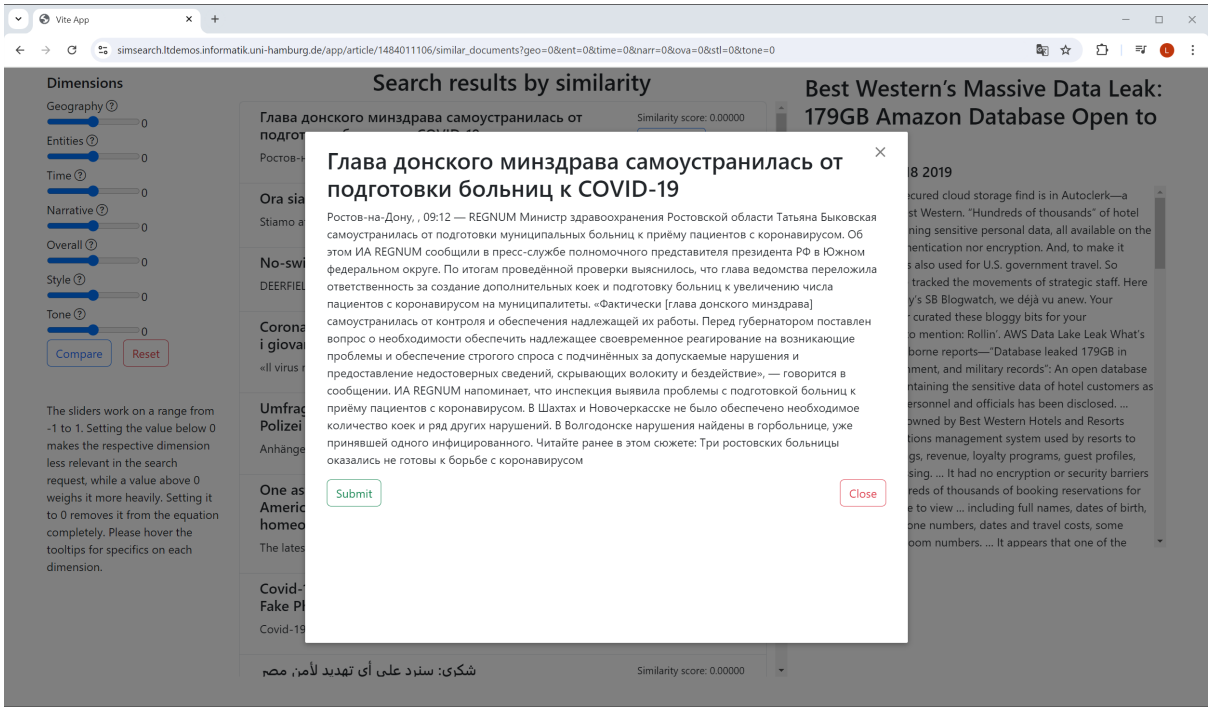


Figure 4.2: Modal for detailed article view

To compare the results we yield from the similarity search, we also provide a more traditional search approach. Users will be alternating between the two during the course of the study. The structure largely follows the same as the similarity search, the difference being in the search bar on the left side. On the front-end the user will find just a search box, each query entered will then be sent to our Elasticsearch back-end and return articles that contain that term, see Figure 4.3. Data from both interfaces will be collected in one MongoDB collection each, storing the search parameters as well as the time taken from initial search query to the submission of a specific article. In this thesis, we do not consider the exact time taken to configure the parameters or enter the search query. This data is then studied in our analysis.

4.3 Method

We pre-compute all of the sentence embeddings for our dataset so the computation time on any search query is kept as low as possible. Each article is, ahead of time, passed to each model representing the dimensions, returning a tensor for every article. These are then stored and will be used to look up the embeddings on each search. Any time a configuration is sent to the

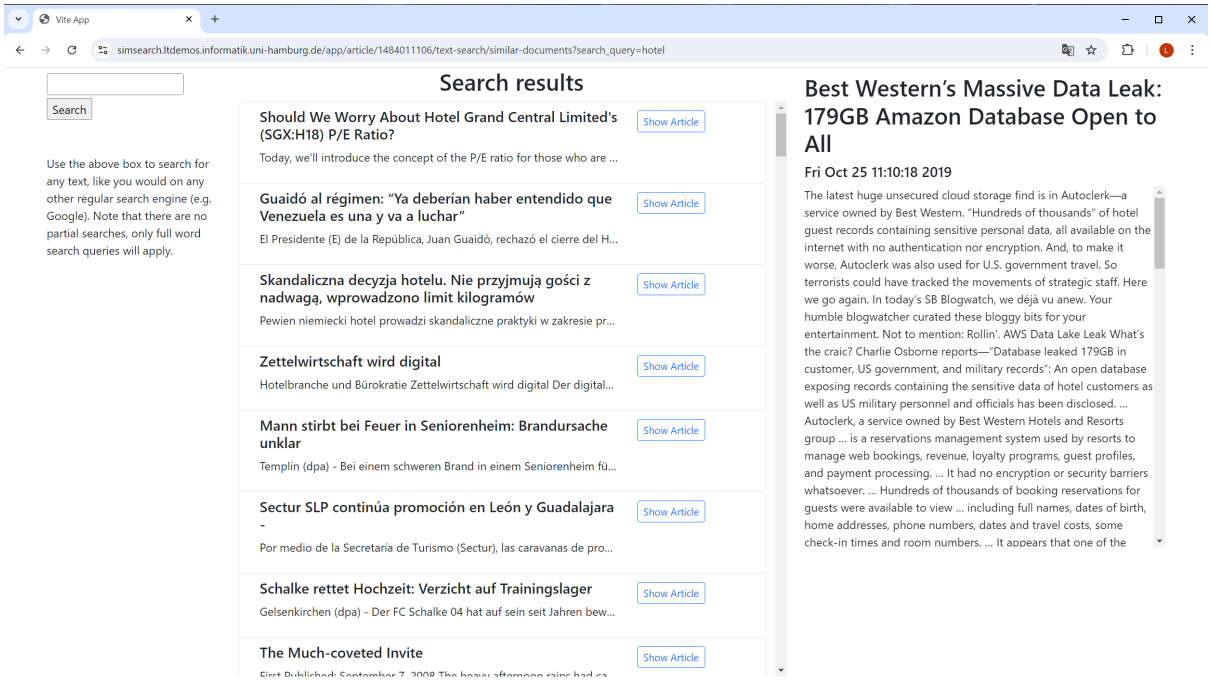


Figure 4.3: Classic text-based search interface

back-end, the cosine-similarity between the query article and each entry in the corpus will be computed. The returned tensors will then be stacked and multiplied by the tensor representing the weights coming in from user input. The resulting tensor will then be searched for the top entries and we pass those entries back to the front-end to display them in the aforementioned results lists. We thought about adding the number of entries that are shown in the results to the user input as well, but decided against it as it brought no practical value for the task. The text search scans all the articles' 'body' fields for the query and returns only those that contain it, using Elasticsearch's standard algorithm for text search. Both of these approaches are wrapped in our application, with which we will then conduct our user experiments. Each participant receives a set of tasks to solve, followed by a survey about their experience with our system. After we finish the study, we will perform an analysis of the data collected in our study with this web-application. For that, we will be looking for certain key factors:

- How much time is necessary to retrieve a relevant target article?
- What similarity configurations are used for any given search?
- How much time is spent on any single task?

- Are the users actually successful in finding a set target article?
- Is there a shared attribute between the wrongfully submitted articles?
- Do users show a bias in preferring to configure specific dimensions? (e.g. entity focus)

Furthermore, we inspect the user survey results to get a general response about the user experience in our experiment, as well as potential feedback and suggestions for improvement. Finally, we will give a conclusion and some thoughts about how our system could be improved to be used in a real-world setting and compete with established search algorithms.

4.4 Experiments

With this setup, we can then start conducting our experiments centered around our problem statement:

Given any news article, please find the ones with the highest similarity from our corpus of articles.

This raises our first question: What would be the time it takes a human to make these comparisons and provide results and can we meaningfully increase this time with our dimension based similarity search? A human would likely normally take this task on with plain text search in common search engines, looking for keywords and somewhat unique identifiers in a given article. The only difficulty there is thinking of shared keywords to search for and filtering unwanted results. To access the question if our approach to a configurable similarity search will improve the search experience we suggest the following hypothesis:

Hypothesis 1: Given a customization option for weighing the potential dimensions in which to compare a set of articles, users will be able to find a set of target articles faster than users on a regular search engine.

To prove this we will set up tasks in an alternating fashion for each participant, with half of the tasks using our search application while the others use a regular text-based algorithm engine, Elasticsearch in our case, for which an interface will be presented. We will then provide each user with the article chosen as the basis for the task. Then we will ask them to find target articles, similar in some of our defined dimensions. During the experiment, we will be taking the time on both groups to evaluate the actual difference in time spent researching. Along with this measurement we will also be looking into the actual search results the groups came up with and if those line up with the expected results.

This then leads to the next question: Why do we need multiple dimensions of similarity? To answer this question, we structure our tasks in a way that certain dimensions will be either very different or very similar. Our hypothesis for this question is:

Hypothesis 2: Any given task has important differences or similarities in multiple dimensions, therefore being able to configure several of them is advantageous

To answer this we closely monitor all the similarity settings that are used throughout each task. In every case, there should be at least two dimensions configured to retrieve the relevant target. The similarity search should also be able to more precisely filter and sort the results, as opposed to the traditional approach that applies no ranking at all, only retrieval.

Finally, we consider the question of the relevancy of our dimensions. Our theory is that given any research task, the participants will likely limit their dimension scales to a certain sub-set of the available ranges. The hypothesis here is the following:

Hypothesis 3: There will be a bigger focus on the entities, time and geography dimensions, since those will probably yield good results for most search queries.

For this we will store the search parameters of all participants to evaluate to which extent the different scales have been used. Then we will compare this across all of the users we have to generate an overview for all dimensions from which we then conclude the general relevancy of the filters.

5 User Study

5.1 Idea

For our studies, we plan to have about 20 participants taking part in our experiments. For each one there will be 6 tasks, split between the classical text-based search and our similarity-based approach. The order of these will be alternating per user. Then each individual user will be guided through the following process: We send each user a link to our study form we set up with Microsoft Forms[1] where we will present them a brief overview of the task. This overview includes an explanation what to do and to expect and some visual introduction in the form of screenshots from each search engine. For the classic approach, they will be faced with a text-based search box and will use keywords to find related documents. In the case of a similarity-based task, they will be using our slider-based interface, in which they configure the weight sliders per dimension and press the search button to browse the top related articles from our corpus. We will add both starting and target articles to the corpus manually to avoid any bias from model training, so the corpus will function as the distractor. A detailed view of the document will be available constantly during the search. After all tasks are completed, the participant will then be asked to take a small survey for our reports, which we will then be using to gather some qualitative analytics of our setup, see Figure 5.1.

To provide an example for a given task, it could be as so: The user will be presented with an article from a reputable source about the demonstrations against right-wing extremism in Hamburg. They will then be asked to find an article from a contrary, possibly extreme point

1. https://forms.office.com/

of view. The target would then be to find an article denying the demonstrations happening in Hamburg.

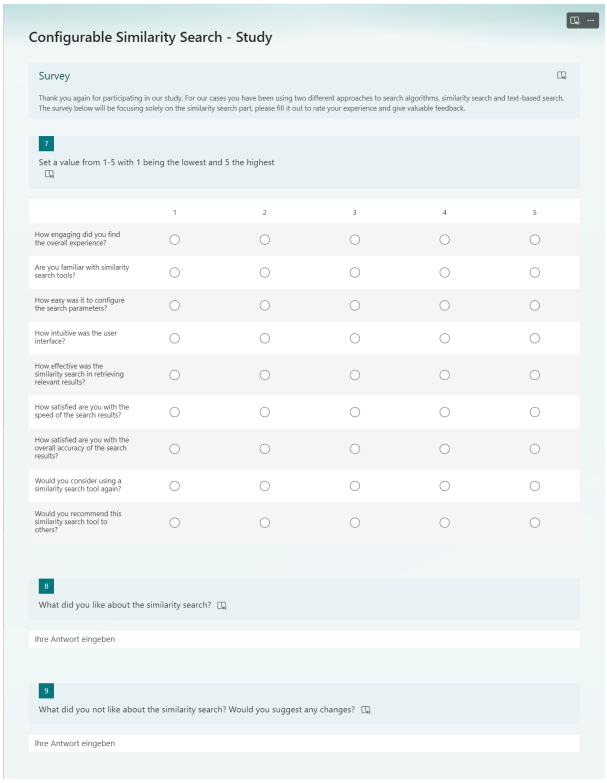


Figure 5.1: User survey about the experience in our study

In each experiment we will be logging the following data: Time it takes to complete the tasks, settings in configurable elements of each type of task and the submitted articles to check for completion as well as correctly identified target articles. The last step of the study takes the participant to our survey, where we collect data about user experience with the interface in general, possible experiences with non-text-based search approaches and happiness with the results received from each similarity search request. The survey focuses solely on our similarity search. Finally, the users will get the opportunity to raise any problems they encountered as well as give feedback for what they liked or disliked about it and how to further improve the interface.

5.2 Tasks

Now we will have a closer look at the actual experiments, to highlight some of the features we wanted to include in our articles to further analyze user interaction at the end, regardless of results (e.g. do users actually read the articles or does time play a factor in decision-making).

Task 1 - Covid-19:

The first task a participant will be confronted with is an article about Boris Johnson being admitted to the hospital with a Covid-19 infection. The user will be asked to "Find another article of a head of state contracting Covid-19". The hypothesis here is that this should be easy enough to solve as the dataset contains plenty of articles about Covid-19 since it was a very important news topic and the time range for the dataset also matches the timeframe of the pandemic. From this subset of articles about Covid-19 there is also a large enough number of reports on state representatives falling ill. This will however also likely be the reason that there will be a lower percentage of 'correct' answers, as the target article we set will not be likely to be submitted as there are plenty other articles that fit the same description/task. However, it can also be presumed that most if not all of the articles submitted will be related to Covid-19 and politicians. We expect users to focus on the overall and tone dimension. Entity focus might also apply here, however there is a mismatch for that dimension as the entity "Covid" is shared, the actual politicians however are different. Arguably time could also be a relevant factor, however the target article was published two years later so it might be misleading.

Task 2 - Hambacher Forst:

For the second task, users will be presented with an article about the occupation of the Hambacher Forst in Germany. The article is written from a more conservative standpoint, not endorsing the actions taken by activists demonstrating the destruction of the Hambacher Forst, with some clear framing of those people as violent criminals. The user is asked to provide "An article contrasting the depiction of brutality in the events". Our target here is an article with accounts of police brutality that some of the demonstrators experienced, published by the left-leaning Die

Tageszeitung (taz). There are two passing articles that we deem correct, however there is another third option which we suppose will be submitted as well because of the wording of the title. We want to use this task to evaluate the actual reading spent during each task. We hypothesize that users will read the title of that third option that reads "Ein Schlag in die Magengrube", a title that could lead readers to assume it was accounts of violence and submit this without further reading of the article body. It is however just a figure of speech, the article itself only talks about the politics behind the Hambacher Forst protests without any accounts of brutality. We think the focus for this search will be on entities, geography and time. The articles are very dissimilar in the narrative, style and tone dimensions, so we would expect users to set negative values in these dimensions for their requests for this task.

Task 3 - European Election:

The third assignment will be centered around election participation in the European Union. The starting text will contain information about participation in Germany in the 2024 election. The task is then to submit an article with information about the previous European election. Here we again have a small set of 3 articles we would deem as the correct answer, however we predict some similar articles about the current election to be submitted as well. All of the articles are taken from official and mostly formal sources e.g. Tagesschau so the obvious focus here is the entity and tone dimension as similar, while the time dimension is very dissimilar. With this we aim to evaluate users' perception of metadata, in this case the publish date. Obviously, articles from 2024 will likely be covering the European election in that year. The targets are mostly from 2019.

Task 4 - UEFA:

The next task focuses on the European Championship 2024. The starting point here is an article from the UEFA itself, detailing how the revenue from the event will help the sport going forward. The user is then asked to "Find an article about the source of the revenue of the European Championship 2024". This is the first of the tasks featuring multiple languages, with the base being English while our targets are all taken from German news outlets. We don't anticipate

many English submissions since most of our participants will be German speaking, however we want to evaluate whether the starting language influences the decisions in any way. Also, there is a discrepancy between base and targets, as the news coverage of this topic generally outlines some very expensive costs for the host cities of the European championship. This information is obviously not present in the article by UEFA. We again have a focus on geography, time and entities as similar, while tone and narrative should be pretty dissimilar.

Task 5 - Wildfires:

The penultimate task will be centered around reports of wildfires across the globe. An obvious implication of this is once again that there will be reports in multiple languages covering similar scenarios. Our selected set will feature a report on wildfires in Australia as the base, with the research question being: "Find another article reporting on wildfires on a different continent". Our target here is a report about the increasing fires in the Amazon rainforest. For this search, the dimensional similarities are not quite as obvious since there are not a lot of overlaps except for the wildfires themselves. We expect some similarity settings for entities and narrative however, while geography should be very dissimilar. The target article is also different in tone from the base, however we don't anticipate our users making this connection without prior knowledge. In the text search, there is also a small hurdle designed with this, as we expect users to search for the term "Brand" as in "Waldbrand" which has completely different meaning in English, so articles from that language should not really be matching that request.

Task 6 - Israel Critics:

Finally, the last set of articles will feature information about Israel and the ongoing situation in Gaza. Both articles are written in a perspective critical of the political and militaristic actions of Israel in the Gaza region. Both reports are written from the perspective of Palestinians from different backgrounds, also condemning the lack of outrage and action towards Israel from the rest of the world. Users are tasked with "Find another article criticizing Israel's politics from the view of Palestinians", so there is a clear set of similarities here we expect to see (geography, narrative, tone). This should also prove relatively easy to solve with the text search as well, so

overall this is one of the easier tasks. The language difference between start (German) and target (English) is probably the biggest difficulty here. Apart from the task-specific observations we aim to gather here, there are some overarching points we will be checking for:

- General return rate of people invited
- Does the time decrease over the duration of the study (fatigue)
- The first similarity search takes the longest (unfamiliarity)
- Percentage of correctly submitted articles (including some unintended similarities)

6 Analysis

In this section, we are going to take a closer look at the data we have collected through the application as well as our user survey and analyze them in regard to our research questions. We will evaluate how well the users did on the overall tasks, what we can learn from the submissions that were not part of our target setup and how each participant felt with our general study. First, we will inspect the overall statistics of the study and then we will take a detailed look at the results for each task.

6.1 Qualitative Analysis & User Experience

For our experiment we sent out 23 invitations to users from different backgrounds and technical experience. Of those we have a total of 14 final submissions to our study form, leaving us with a response rate of 60.9%. Another three of those submissions did either not submit any article or only did a part of the experiment, so we have omitted those entries from the actual results analysis for better clarity and robustness of the data. Through our survey we have collected feedback about the user experience and existing knowledge about similar technology, see Figure 6.1. Generally the overall experience received favorable reviews, with only 28.6% placing it in the lower section on our Likert scale. The fact that similarity search tools are not commonly found in real world scenarios is reflected in that data, as 42.9% of users reported no experience, while 35.7% started with a good understanding of the technology. These groups can easily be split into the friends and family that participated and the invitees from our university respectively. This suggests a similar distribution for the first task of the experiment as well, as those that are unfamiliar likely take longer to adjust. Furthermore, a lack of explanation for the dimensions in

the similarity search tool is the most described critic from the feedback we gathered. This is in relation to their actual effect on the search as well as differences between seemingly analogous dimensions.

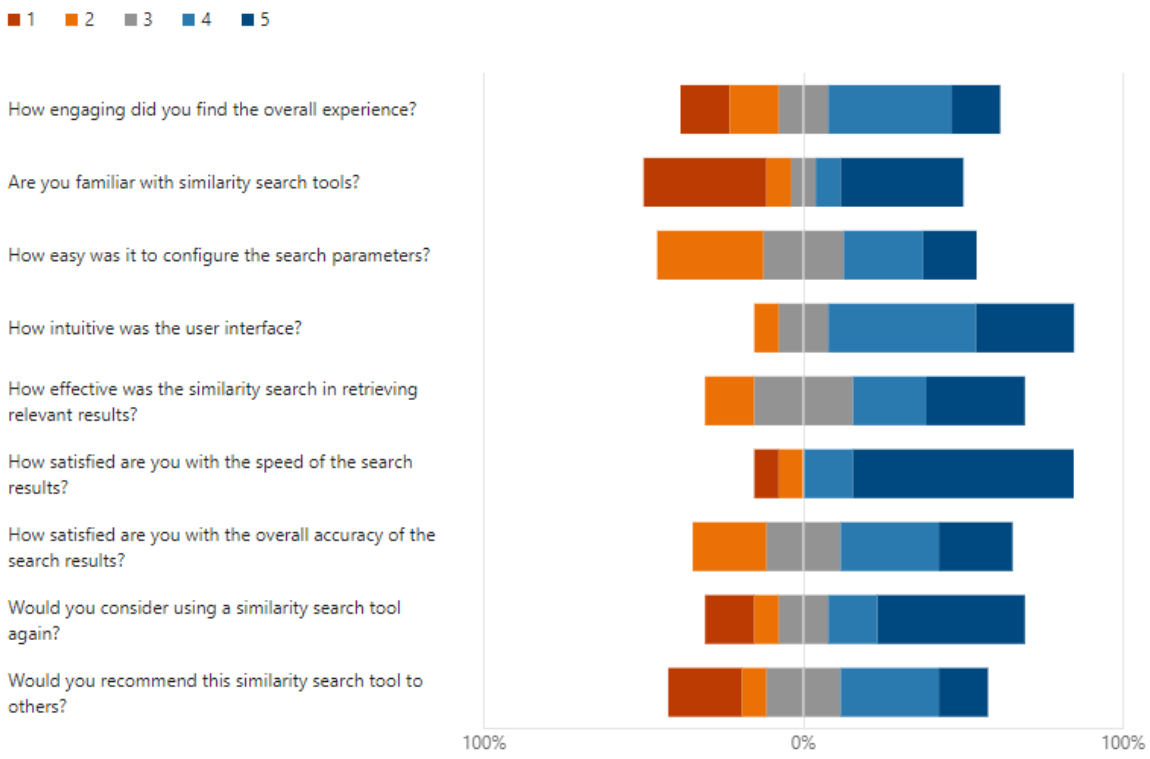


Figure 6.1: Likert-scale representation of the responses to our user experience. Ranking from 1 (lowest) to 5 (highest)

We see this sentiment again in the responses to questions about the intuitiveness and ease of use of the similarity search tool. While 78.6% said the user interface was intuitive, only 38.5% also found the parameters easy to configure. This further underlines the feedback from the free text responses as mentioned above. In terms of effectiveness, our application also did well, with almost all of the participants being very satisfied with the speed of the search results and only roughly 20% being dissatisfied about the accuracy of the results. This likely ties in with the overall problems with the dimension settings. The questions about future use show that people are willing to try other approaches to searching data generally, as only 21.4% of the participants would not consider using a similarity search tool again. The number of users who would recommend our application is somewhat lower, with 57.1% not likely to suggest its use. We explain this difference in numbers with the problems our users faced in configuring search

requests with the similarity sliders and the explanations thereof. Improvements to this could include an even more detailed instruction before the task section of the experiment, possibly providing concrete examples of what is to do and what to expect from a given query, as well as more detailed tooltips. Some users were also confused by the presence of articles that are not in German or English, which begs the question if our multilingual setup is of much value in our scenario, as while those do work as distractors in the dataset, they are never actually respected in any search task. Even if there were similar articles in the results, users would not know that if they do not speak the respective language.

To summarize, the study process was generally successful, with a decent completion rate and positive feedback, the mainly reported issues being concentrated on problems with information regarding the dimensions. We will now have a look at the quantitative data returned from our study and analyze the technical effectiveness of our application.

6.2 Task Evaluation

To evaluate our study we are going to inspect our data in relation to our research questions and see to what extent we were able to answer these. First off we are going to check our submissions for their success rate with each search type. In Table 6.1 we can see all entries we received based on three categories:

- Target: Article is one of our pre-selected target articles
- Similar: Article is not in our target articles but still solves the task correctly; Regarded as a success
- Wrong: Article did not solve the task correctly / is not in our targets

There are a total of 11 entries per article so to get our overall success rate we calculate the percentage of successful attempts across all 66 tries. With 45 correct attempts this leaves us with a rate of 68.18% for the overall search submissions. For the similarity search we get 72.73% and

	Similarity Search			Text Search		
	Target	**Similar**	**Wrong**	**Target**	**Similar**	**Wrong**
T1 - Covid-19	0	3	1	3	2	2
T2 - Hambacher Forst	0	0	7	0	0	4
T3 - European election	3	1	0	6	0	1
T4 - UEFA	7	0	0	4	0	0
T5 - Wildfires	2	2	0	2	0	5
T6 - Israel critics	2	4	1	3	1	0

Table 6.1: Distribution of answers per article

the text search ranks lower at 63.64%. This suggests that the similarity search actually performed better in retrieving correct results than the classical approach. Another observation here is that the similarity search provided more results classified as 'Similar', whereas the text search seems to return more clearly distinguishable results. We are going to conduct a Chi-squared test for independence to check for statistical significance of our results, where we will group 'Target' and 'Similar' categories together as a success in our contingency table. Our hypothesis is that there is a dependency between success rate and the respective group. We see the observed values again in table 6.1 and calculate our expected values from our contingency table 6.2. This is calculated by $\frac{(row_t * column_t)}{total}$ with t = total respectively. From this we get an expected value of 22.5 for success and 10.5 for fail in both cases, as the row and column totals can not differ between the groups. Now we can calculate our χ^2 with the formula:

$$\chi^2 = \sum_{k=1}^{n} = \frac{(observed_k - expected_k)^2}{expected_k}$$

With this we get $\chi^2 = \frac{(24-22.5)^2}{22.5} + \frac{(21-22.5)^2}{22.5} + \frac{(9-10.5)^2}{10.5} + \frac{(12-10.5)^2}{10.5} = 0.214$ Our degree of freedom is df = $(rows-1)(columns-1) = 1$ We will use the standard significance level of 5%, so our p-value is 3.841. Since our chi squared value is smaller than our p-value, there is no

statistical significance here, so our hypothesis that there is a dependency between the groups and their success is denied.

	Similarity	**Text**	**Total**
Success	24	21	45
Fail	9	12	21
Total	33	33	66

Table 6.2: Contingency table for our Chi-squared test with 2 categories

Now we will run the test again, this time also taking into consideration all 3 categories. We have our contingency table in table 6.3. Our respective expected values are 16 for 'Target', 6.5 for 'Similar' and 10.5 for 'Wrong'. This leads to $\chi^2 = \frac{(14-16)^2}{16} + \frac{(18-16)^2}{16} + \frac{(10-6.5)^2}{6.5} + \frac{(3-6.5)^2}{6.5} + \frac{(9-10.5)^2}{10.5} + \frac{(12-10.5)^2}{10.5} = 4.698$. Our degree of freedom here is df = $(rows-1)(columns-1) = 2$. Again we use the standard significance level of 5%, resulting in a p-value of 5.991. Unfortunately our calculated chi square value is still below the p-value so our hypothesis is also denied for all 3 categories.

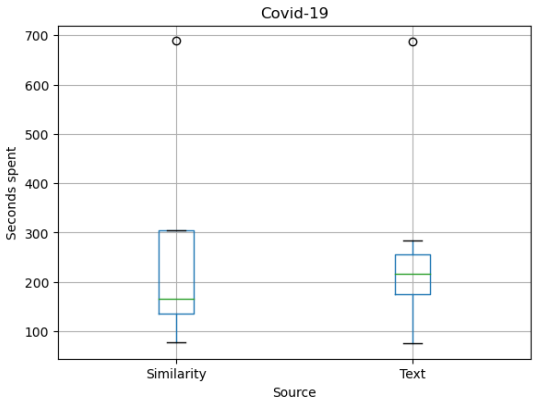
	Similarity	**Text**	**Total**
Target	14	18	32
Similar	10	3	13
Wrong	9	12	21
Total	33	33	66

Table 6.3: Contingency table for our Chi-squared test with 3 categories

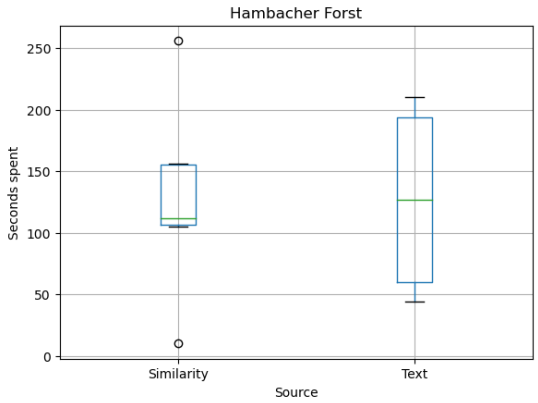
Next, we will take a closer look at the individual articles to see how the performance in regards to time and correctness is on each of them and also take into consideration the hurdles we set up in our experiment (e.g. 'trap' article for Hambacher Forst).

In Figure 6.2a we can see the time spent per user for the search on the Covid-19 infection article. While there are some extreme outliers here, likely because this is the first task for all of the participants and there is some time needed to get familiar with the interface, we still get an

impression of the speed and precision of our approaches. Since the outliers are also roughly equal in both types of search, the averages we get here are still valid, the similarity search taking users about 3 minutes, while the text search is slightly under 4 minutes. The success rate on the similarity search is 75% for this task, while the text search only achieves 71.43% so for the first task the similarity search outperformed the classical approach in both speed and success. Another notable observation here is that the Covid-19 articles had the widest range of successful articles, as the dataset contained a high number of reports about Covid-19 infections in politics.



(a) Time spent on the Covid-19 article in seconds



(b) Time spent on the Hambacher Forst article in seconds

Figure 6.2: Box plots for task 1 & 2

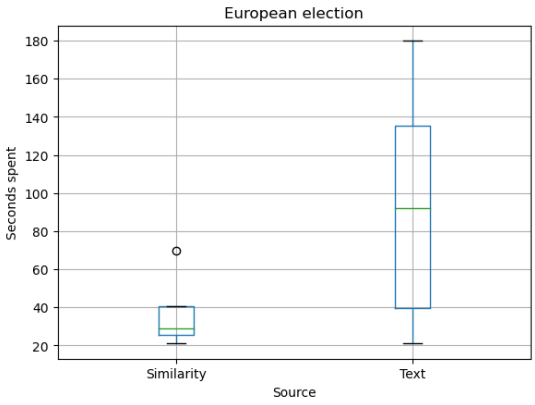
Next up we have the comparison on the article about brutality in the Hambacher Forst, see Figure 6.2b. Generally, the overall time spent on this task is below the first task, an observation that will also be true for the following tasks. This is either due to users getting comfortable with our application or signs of early fatigue, as in the users want to get the experiment done quickly. Another factor might be that the dataset contains much fewer articles about this and the following topics, as opposed to the Covid-19 related articles. In terms of time, the median value for the similarity search is slightly under 2 minutes and the text search just above the 2 minute mark. Success rates are 0 for both approaches, given the fact that we set a very strict rule of classifying articles as 'Similar'. We implemented this as a test to see if users would actually read an article before submission or just select an article based on the title and/or some keywords. We can confirm this as >90% of the submissions were closely related to the topic but did not actually contain any accounts of police brutality, which was what the task asked for. The

remaining results were regular failures, with no relation to the task at hand. If we dismiss the strict rule, the distribution would change as follows: 7 'Similar' articles for the similarity search and 2 'Similar' plus 2 'Wrong' articles for the text search, leaving us at 100% and 50% success rate respectively. All of the articles now classified as 'Similar' contain information about at least the Hambacher Forst in some way, while 2 of the 4 submissions in the text search report on completely different topics.

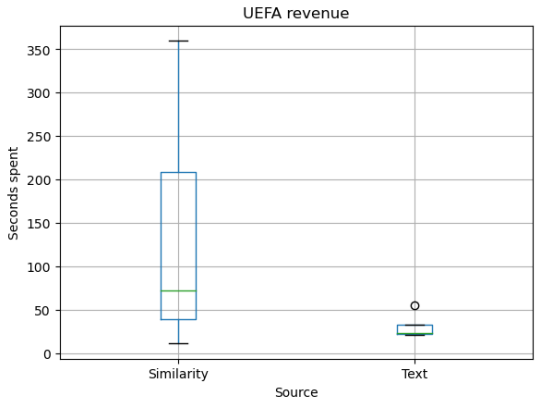
The next task to discuss is the European election participation. The main factor we wanted to test for here is the difference in the time dimension, expecting users to submit an article from 2019 when the last European election took place. There were however some articles in the dataset dated in 2024 that also fulfilled the task properly, resulting in high success rates. For the similarity search we get a rate of 100% with no 'Wrong' articles submitted, while the text search did get one entry that was the exact same article as the task provided. This is a fault on our end, as there should not have been a duplicate article in the dataset, however the task was to find a different article so we did not accept the submission as similar. With this in mind, the success rate is at 85.71%, it is however likely to also land at 100% given we had discovered the duplicate sooner. In terms of speed, the similarity search seemingly outperformed the text search by a large margin, its top-end outlier being around 70 seconds, which is still faster than the median of the text search at around 93 seconds. Since the results are basically 100% correct, given the exception mentioned earlier and also very fast in an overall comparison, this was likely an easier task. The dataset was favorable here, as like we mentioned before, there were several articles solving the task outside of the target articles.

Moving on we will now look at the UEFA related task. The idea here is mainly to find an article in another language (German) since the starting article is published in English. While this was also true for the first task, there we looked for articles that were in the same language. Other than that we were looking for a difference in tone and narrative in our target articles, as those were all critical of the costs of the European Championship. The results show that this task was very clear, as all entries for both search engines were in our target articles and therefore correct. The time spent on this task is also notable, with most users at or below 1 minute and

only two outliers above 5 minutes in the similarity search, still achieving a median of just over 1 minute. For the text search this went even faster, with the median here being at 30 seconds. In terms of the data, there are several other articles related to football as well as the UEFA in the dataset, however a big percentage of those is not in German, so the targets were somewhat obvious. Generally however this use case seemed suitable for our application.



(a) Time spent on the European election article in seconds

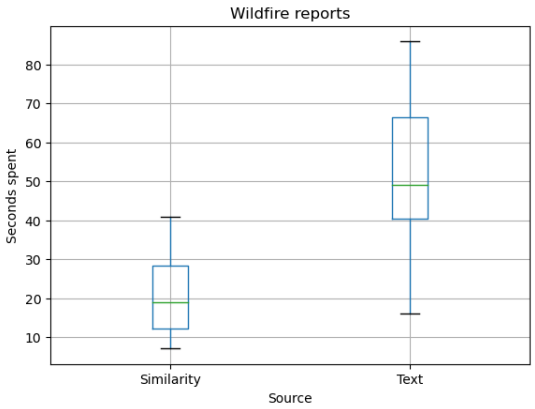


(b) Time spent on the UEFA revenue article in seconds

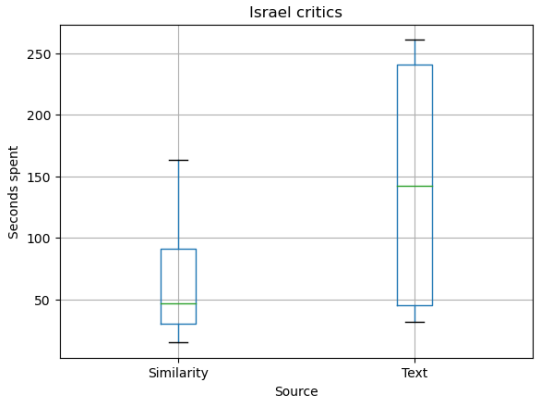
Figure 6.3: Box plots for tasks 3 & 4

Now we will analyze the results of the fifth task, asking users to find an article about wildfires on a different continent. The idea we wanted to implement here is a hurdle mainly applied to the text search, to check if this would make a difference in the results. To achieve this, we tried putting together a task involving a play on words, presenting the user with a German article about wildfires in Australia, asking for another article about wildfires in a different continent. We thought German users would likely try to search for 'Brand' which has a different meaning in English and also does not really result in any meaningful articles for the task. The same goes for 'Waldbrand', which does not have a separate meaning in English, but does not return correct results either. To summarize this, the task was not as obviously solvable with the text search as other tasks. With the similarity search however, this task was solvable more easily with a setting that is dissimilar in the geography but similar in entities and/or overall. The results we gathered support our idea for this task, as the participants using the similarity search for this task were a lot more successful, with a 100% success rate, while the text search users only managed to get 28.57% with two out of seven correct answers, see table 6.1. The actual search requests for

the text search further underline this, as most of the queries were actually in German instead of English. This supports the idea that most users tend to query the search depending on the language of the starting article or the task. For English based articles, queries were mostly English and vice-versa for German. The time spent on this task, as seen in Figure 6.4, also reflects the gap in difficulty between the two engines. While both groups solved the task very fast, the median for the similarity search is just under 20 seconds, with the text search median being more than double that time at close to 50 seconds. Generally however, only two users spent over 1 minute on this task. Since this is the penultimate task, there might be some study fatigue setting in with users who then submit vaguely related articles to finish with the study quickly. We will see if this also holds true for the final task. The hypothesis we set up at the beginning of this task seems to be fulfilled, given the data we just analyzed.



(a) Time spent on the wildfire article in seconds



(b) Time spent on the Israel critics article in seconds

Figure 6.4: Box plots for tasks 5 & 6

We will now look at the statistics for our final task of the study. Since this is the last step of the study, we did not add a lot of hidden tests to this and aimed for an easier task. The starting article is written by 3 Palestinian activists criticizing the geopolitical actions of Israel and the lack of actions by the countries of the West. Users are tasked to find another article on the same topic that is also clearly written by a Palestinian. In both cases the targets should be easily acquired, a simple search for 'Israel' on the text search or a geographical similarity on the similarity search will yield the correct article among others. The results show that finding the correct article was in fact that simple, with only one answer in the similarity search not in the 'Target' or 'Similar'

categories and 100% success rate for the text search. The time spent on the task supports this as well, with the median for the similarity search (which had the majority of participants for this task) being around 50 seconds. For the text search the median is over 2 minutes, however looking at the time in detail will show that there were two extreme values (>4 minutes) and two answers under one minute. Since the success rates are high it is difficult to bring up the topic of study fatigue again here. We would have expected time spent to go down along with an increase in 'Wrong' articles and while the time did in fact decrease, the success rate did not drop as much. From this we can not draw a direct conclusion, as users might have wanted to quickly finish the study and just submitted correct articles by accident, or they were just able to find the correct articles in that short time frame.

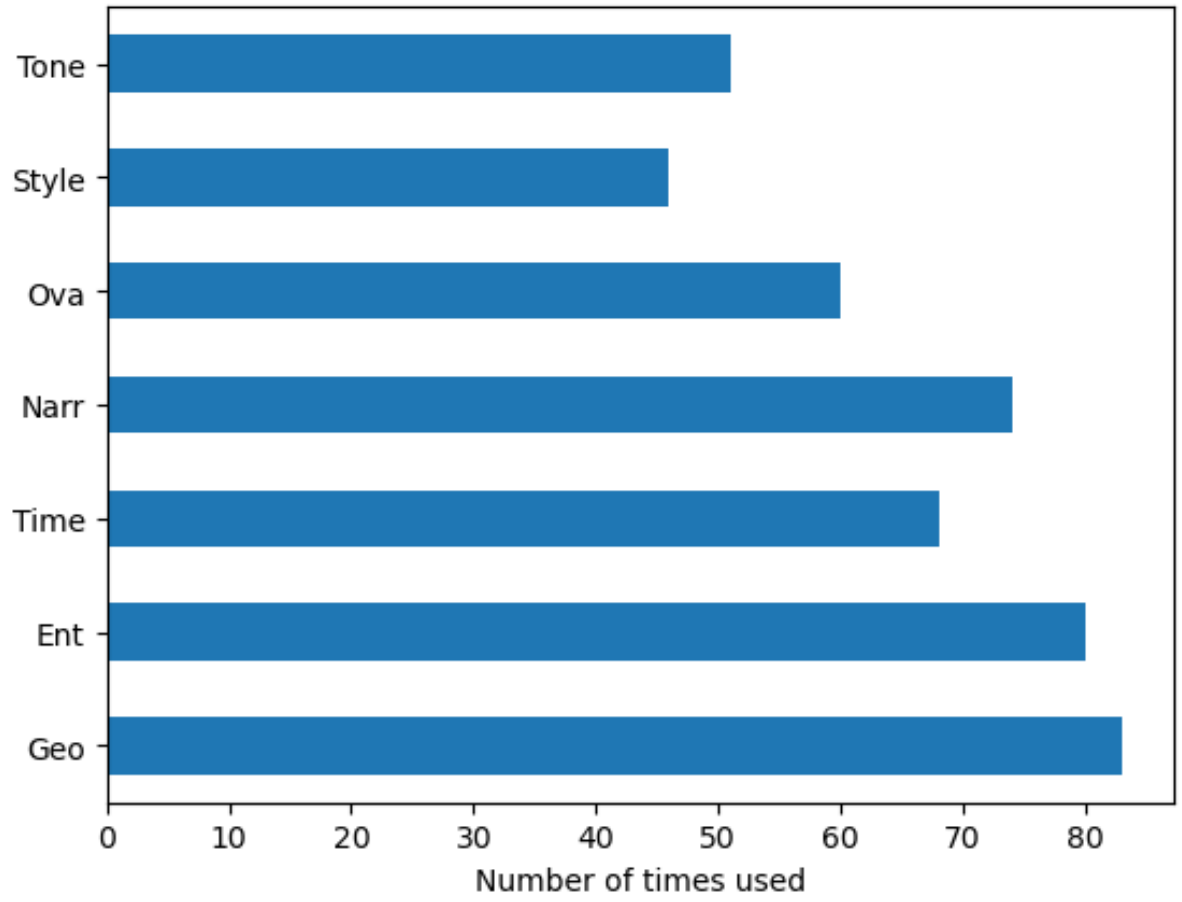


Figure 6.5: Bar plot of all dimensions of similarity used through all entries

6.3 Dimensions

In the last section of our analysis, we want to investigate the dimensions that were used some more, also taking into consideration some of the feedback of our survey. In Figure 6.5 we see the distribution of each dimension across all queries. In the beginning of this paper we suggested that there will be a focus on some of the more 'clear' dimensions, like entities or geography, for which any participant will have a tangible understanding right away. On the other hand this means that some less obvious dimensions will be used less, mostly because of the unclear effect. Another factor to this however are the actual tasks, which ideally require different dimensions of similarity to be solved. 'Geography' was the most used category over all tasks, even slightly ahead of 'Entities'. Since there were 4 tasks for which the geography was somehow relevant, it makes sense that it ranks this high. The 'Entities' dimension should most likely always be configured in our use case, as news articles usually report on some form of entity which is either similar or dissimilar to what we are looking for. 'Narrative' being used more than 'Time' is also in accord with our tasks, as 3 of those clearly asked for some form of narrative in the article. Seeing the 'Time' dimension with close to 70 usages is interesting, as only one article really asked for a different time from the starting article. Similarly, the remaining 3 categories of 'Overall', 'Style' and 'Tone', while at the last 3 ranks, still have a decent amount of usages. Upon closer inspection of the actual data we explain this with the fact that some of our participants always set all dimensions to some degree, as opposed to picking specific dimensions to look for per query. Generally however these dimensions seemed less useful overall and are probably not required to solve the same tasks in a similar manner, especially in the case of poorly explained effects.

7 Conclusion and Future Work

To summarize our results we can say that even though we were not able to prove a statistical significance, the data we gathered still provided valuable insight on the general possibility and usability of similarity search interfaces. Our data suggests that users can get familiar with a new way to browse data fairly quickly. Furthermore, the actual results obtained from our new search engine propose that, at least for the task structure we provided, users can retrieve relevant articles and solve the tasks in a similar manner to a classical search approach. Considering that most users never used a similarity type search before, the overall speed statistics showed that it was actually faster than the text search for almost all tasks, when taking into consideration all outliers. Success rates also seem favorable for the similarity search. There is however still much room for improvement for our approach. The first step would be to get a bigger data sample to obtain significant results, to properly prove the hypotheses we formulated in this paper. A closer look at the dimensions can also improve our setup significantly, by not only going into further detail on each of them and their respective effects, but also possibly removing some of the underperforming dimensions and maybe even think of new ones to add. Generally however, the dimensions that we did implement in our scenario worked as intended. Our initial question about the need for multiple dimensions and which specific dimensions to use is strongly tied to the scenario in which the similarity search will be used, as different applications of this concept will require different dimensions. This also applies to the multilingual dataset, which is a good proof of concept for this thesis to show that the models actually perform similarity search across several different languages. In a real world scenario however, this is not really applicable, as it is very unlikely for users to actually understand all of the languages and gain any information from those articles.

For example, as suggested by Hatzel et al. (2023), a possible future use case would be to move away from news articles and instead construct a similarity search for literature. In this case, other dimensions could prove much more valuable, e.g. 'Style' and 'Tone' could prove to be helpful in a literary context, where users try to find poems or novels with similar styles of writing. 'Geography' or 'Time' on the other hand would likely be less useful here, as those would be only scanning the contents of any given text for similarities. They also suggested a scenario for product reviews, in which they used a heavily weighted 'Tone' dimension to retrieve product reviews of a certain rating, or a model that focuses on 'Overall' and 'Entities' from which they subtract the 'Tone' dimension to achieve the inverted effect of finding products with different ratings. Furthermore however, a similarity search could also be used outside of the written text domain. In pop culture for example, models could be trained to reflect certain dimensions of music, from which an interface could be built to filter large music collections (e.g. Spotify) by those dimensions instead of usual artist and song names. Another idea could be to train models on certain dimensions for video games, e.g. game mechanics, story ideas or technical specifics. Users could then use this to find more video games that are similar in those aspects, as this is a task that is usually hard to identify before playing. In any case, the actual set of dimensions to be used as well as a clear explanation and effect are of the biggest importance for any possible application of this concept. Human understanding is not a limiting factor here, as the entry hurdle for using different approaches to search engines seems to be low. With this work, we hope to contribute not only to the field of news article similarity but also to semantic similarity and new approaches to search engines in general and maybe inspire others to pursue and explore similar ideas.

Bibliography

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, and Ray Kurzweil. 2018. Universal Sentence Encoder, arXiv:1803.11175, April 12, 2018. arXiv: 1803.11175[cs].

Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining,* 785–794. KDD '16. New York, NY, USA: Association for Computing Machinery, August 13, 2016.

Xi Chen, Ali Zeynali, Chico Camargo, Fabian Flöck, Devin Gaffney, Przemyslaw Grabowicz, Scott Hale, David Jurgens, and Mattia Samory. 2022. SemEval-2022 Task 8: Multilingual news article similarity. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022),* 1094–1106. SemEval 2022. Seattle, United States: Association for Computational Linguistics, July.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. arXiv:1810.04805. arXiv, May 24, 2019.

Liat Ein Dor, Yosi Mass, Alon Halfon, Elad Venezian, Ilya Shnayderman, Ranit Aharonov, and Noam Slonim. 2018. Learning Thematic Similarity Metric from Article Sections Using Triplet Networks. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers),* edited by Iryna Gurevych and Yusuke Miyao, 49–54. ACL 2018. Melbourne, Australia: Association for Computational Linguistics, July.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022. Language-agnostic BERT Sentence Embedding, arXiv:2007.01852, March 8, 2022. arXiv: 2007.01852[cs].

Felix Hamborg, Norman Meuschke, Corinna Breitinger, and Bela Gipp. 2017. news-please: A Generic News Crawler and Extractor. In *Proceedings of the 15th International Symposium of Information Science,* 218–223. Berlin, March.

Hans Ole Hatzel, Fynn Petersen-Frey, Tim Fischer, and Chris Biemann. 2023. Dimensions of Similarity: Towards Interpretable Dimension-Based Text Similarity. In *Frontiers in Artificial Intelligence and Applications,* edited by Kobi Gal, Ann Nowé, Grzegorz J. Nalepa, Roy Fairstein, and Roxana Rădulescu. IOS Press, September 28, 2023.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. arXiv:1908.10084. arXiv, August 27, 2019.

Amit Singhal. 2001. Modern Information Retrieval: A Brief Overview.

Derya Soydaner. 2022. Attention Mechanism in Neural Networks: Where it Comes and Where it Goes. *Neural Computing and Applications* 34, no. 16 (August): 13371–13385. arXiv: 2204.13154[cs].

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention Is All You Need, arXiv:1706.03762, August 1, 2023. arXiv: 1706.03762[cs].

Eidesstattliche Erklärung

Hiermit versichere ich an Eides statt, dass ich die vorliegende Arbeit im Bachelorstudiengang Software-System-Entwicklung selbstständig verfasst und keine anderen als die angegebenen Hilfsmittel – insbesondere keine im Quellenverzeichnis nicht benannten Internet-Quellen – benutzt habe. Alle Stellen, die wörtlich oder sinngemäß aus Veröffentlichungen entnommen wurden, sind als solche kenntlich gemacht. Ich versichere weiterhin, dass ich die Arbeit vorher nicht in einem anderen Prüfungsverfahren eingereicht habe.

Unterschrift: *Ort, Datum:*

Erklärung zur Veröffentlichung

Ich stimme der Einstellung der Arbeit in die Bibliothek des Fachbereichs Informatik zu.

Unterschrift: *Ort, Datum:*