Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
Pfaffenwaldring 5B
D-70569 Stuttgart

Master Thesis

Evaluating Methods of Improving the Distribution of Data across Users in a Corpus of Tweets

Milan Milovanović

Studiengang: M.Sc. Computational Linguistics

Examiners: Prof. Dr. Sabine Schulte im Walde
Dr. Michael Roth
Supervisor: Dr. Filip Miletić

Start of Thesis: 15.02.2023
End of Thesis: 15.08.2023

Erklärung (Statement of Authorship)

Hiermit erkläre ich, dass ich die vorliegende Arbeit selbstständig verfasst habe und dabei keine andere als die angegebene Literatur verwendet habe. Alle Zitate und sinngemäßen Entlehnungen sind als solche unter genauer Angabe der Quelle gekennzeichnet. Die eingereichte Arbeit ist weder vollständig noch in wesentlichen Teilen Gegenstand eines anderen Prüfungsverfahrens gewesen. Sie ist weder vollständig noch in Teilen bereits veröffentlicht. Die beigefügte elektronische Version stimmt mit dem Druckexemplar überein.[1]

(Milan Milovanović)

[1]Non-binding translation for convenience: This thesis is the result of my own independent work, and any material from work of others which is used either verbatim or indirectly in the text is credited to the author including details about the exact source in the text. This work has not been part of any other previous examination, neither completely nor in parts. It has neither completely nor partially been published before. The submitted electronic version is identical to this print version.

Abstract

Corpora created from social network data often serve as the data source for tasks in natural language processing. Compared to other, more standardized corpora, social media corpora have idiosyncratic properties due to the fact that they consist of user-generated comments. These are, for example, the unbalanced distribution of the respective comments, a generally lower linguistic quality, and an inherently unstructured and noisy nature. Using a Twitter-generated corpus, I will investigate to what extent the unbalanced distribution of the data has an influence on two downstream tasks, relying on word embeddings. Word embeddings are a ubiquitous and frequently used concept in the field of natural language processing. The most common models are often the means to obtain semantic information about words and their usage by representing the words in an abstract word vector space. The basic idea is that semantically similar words in the mapped vector space have similar vectors. In doing so, these vectors serve as input for standard downstream tasks such as word similarity and semantic change detection. One of the most common models in current research is the use of word2vec, and more specifically, the Skip-gram architecture of this model. The Skip-gram architecture attempts to predict the surrounding words based on the current word. The data on which this architecture is trained greatly influences the resulting word vectors. In the context of this work, however, no significant improvement in the results to a fully preprocessed corpus could be found when filtering methods, widely used in the literature, without specific motivation, are used to select a subset of data according to defined criteria, neither for word similarity nor for semantic change detection. However, comparable results could be achieved with some filters, although the resulting models were trained using significantly fewer tokens as input.

Contents

1 Introduction

In the current fast-paced digital age, social media platforms have become a central hub for exchanging information, ideas, and opinions. Twitter, as one of the most influential platforms, plays a pivotal role in shaping public discourse and disseminating news. Twitter [2], which was created in 2006, presents a free service for users allowing them to spread messages up to 280 characters. There are more than 300 million monthly users on the site, who post more than 500 million tweets, essentially comments, per day (Kabakus and Kara, 2017). The massive amount of user-generated content on Twitter has made it an invaluable resource for marketers and researchers alike. The immense popularity of Twitter (or other social media platforms) in recent decades has led to the creation of datasets from tweets for several activities in the field of natural language processing, such as topics in sentiment analysis (Saif et al., 2014) or opinion mining (Pak and Paroubek, 2010). Analyzing such extensive data sets requires careful consideration of the distribution and sampling of the original data to ensure that the insights are representative and unbiased (Bao et al., 2014).

However, before embarking on the analysis, the first crucial step involves the creation of a well-defined corpus, which serves as a representation of data relevant to the chosen research area. The previously mentioned fields are only a fragment of scientific interests based on social network data. The creation of the corpora always depends on certain guiding principles in these fields (Bao et al., 2014; Klein et al., 2021; Pak and Paroubek, 2010). In general terms, this refers to the fact that corpora are not created in vain, but with a phenomenon or objective in mind that is to be studied. In this respect, guidelines, principles, and algorithms are defined which have the respective corpus as a result. The exact form of these pipelines is not crucial for this thesis, but they often involve the same three steps: (i) searching for relevant information by crawling social media sites, (ii) downloading the information and (iii) filtering the resulting corpora on the basis of previously defined criteria (Kabakus and Kara, 2017; Miletic et al., 2020; Tan et al., 2015; Şahinuç and Toraman, 2021).

[2] rebranded to X Corp at the time of writing this thesis

Often one has to remove unwanted data such as spam or near-duplicate content from these corpora in preprocessing (Bao et al., 2014). But even after cleaning malicious and other unwanted content, properties can still be found in user-generated corpora that could be problematic, especially from a linguistic point of view. One of these phenomena is the fact that these corpora are not necessarily equally distributed in terms of the distribution of tweets (Al Sharou et al., 2021). A large part of the tweets are written by a few, but nevertheless highly active users, while the other large part of the users writes very few tweets resulting in a head/tail distribution of content (Sastry, 2012). The distribution, which strictly follows Zipf's law, can be found in several corpora based on user-generated data (Sastry, 2012; Miletic et al., 2020; Klein et al., 2021). The most common solution to counteract the unequal distribution is to subsample these users or their tweets and thus modify the source corpus (Moreno-Ortiz and García-Gámez, 2023; Shoemark et al., 2019).

Selecting a different sample of the data in a corpus of tweets can significantly impact the generalisability and reliability of findings found in the data. Uneven representation of users may lead to skewed results, hindering the ability to draw meaningful conclusions and obscuring essential patterns and trends. Moreover, this disparity can introduce potential biases and misrepresentations, which can have severe implications, especially when dealing with sensitive topics or societal issues (Zhao et al., 2022; Bian et al., 2008).

This is due to the fact that a textual corpus always contains statistical properties in the form of words and their usage. The most important foundation for further tasks in the field of natural language processing is to transfer the semantic information and structures reflected by the words into a representation that can be used to conduct experiments based on this information. A common representation of the statistical properties of any given textual structure is the usage of a word vector space (Schütze, 1993). The underlying idea behind terms in this vector space is the fact that terms that share a similar meaning are close to each other in the space while terms that have foreign meanings are much larger apart. Such a representation

can be achieved by applying word embedding method to the data, which mathematically represents each word as a corresponding dense vector in the word vector space. Word embeddings have been used in a wide variety of applications in the field of natural language processing, such as sentiment analysis (Saif et al., 2014), semantic change detection (Schlechtweg et al., 2019) or word similarity (Elekes et al., 2017) amongst others.

By truncating data in corpora to counteract the problems of user-generated corpora described above, the properties and statistics of their contents are also inevitably changed. This has a particular impact on word frequency and the use of words. A common example of this is users who use a particular expression frequently, and by removing or subsampling these users, certain linguistic information about the terms or their use can be lost to the now less frequent expression. On the other hand, the disproportionate use of certain terms, language patterns, or jargon by highly active users is a fundamental problem in the analysis of corpora as they can disproportionately influence the overall word frequencies with their idiosyncratic language patterns and specific vocabulary choices.

Ideally, word embedding models should be trained on the most optimal version of the data to avoid suboptimal meaning representations due to a selection of a skewed or biased subsample of the original data. The objective is hereby to create a general representation of word meaning of the corpus without being influenced by the top percentage of users that use the words in the vocabulary and skew their latent semantic properties. In a nutshell, one wants to recognize the global representation of all word meanings in the data and not the one of the most active users. Previous studies relying on Twitter data recognize this issue but often use ad-hoc filters without assessing them systematically. (Doval et al., 2020; Tan et al., 2015; Moreno-Ortiz and García-Gámez, 2023)

The aim of this thesis is therefore to explore and evaluate various methods to alter the distribution of data across users in a corpus of tweets. By doing so, I intend to

address the challenges posed by imbalances of data and promote a more equitable representation of users in the dataset by e.g. skewing the balance towards a more evenly distribution and analyzing the aftermath of the changes. In addition, I will use structural and systematic information from Twitter corpora to evaluate the effectiveness of filters that address different aspects of tweets. In this regard, information regarding the length of a tweet based on the number of tokens and information regarding the date on which the tweet was written. To achieve this, I use a standard and widely used word embedding model, namely Skip-gram with negative sampling (Mikolov et al., 2013a;b) to produce word vectors that learn contextual information based on the word frequencies of the respective filtered corpora. The initial data for this thesis is based on a Twitter corpus created by Miletic et al. (2020). This corpus was created with the goal in mind to investigate contact-induced semantic shifts in Quebec English (Miletic et al., 2021) and serves as a starting point for further methods and evaluations of the methods for this thesis. I will then apply the learned representations of the words for all filtered corpora to two suitable downstream tasks, word similarity and semantic change detection.

The selection of the two tasks, namely word similarity and semantic change detection, is explained by the fact that they both rely on the same methodology. This methodology consists of creating vector representations of words, comparing the vectors, and a subsequent creation of a ranked list of the best results, which allows for computationally easy comparison of the results across models and allows for a general evaluation of how well the models understand the semantic details the corpora provide. Therefore, it becomes apparent how effective the selected filtering methods are.

1.1 Objective of this Thesis

This thesis aims to evaluate different filtering methods that aim to improve the distribution of content across users in a corpus of tweets. There are two ways to address this challenge. On the one hand, the distribution of content across users is taken into account by setting up a limit of the maximally allowed amount of tweets per user. Shifting this boundary and randomly subsampling tweets from users that have posted more than the limit, serves as the starting point of this thesis which aims to evaluate different methods of improving the distribution of content. On the other hand, the properties regarding the length and the date of tweets are analyzed in a controlled manner by only selecting a specific subset of tweets depending on their length and the year they were written. In addition, multiple combinations of all three different filtering categories will be considered. The main hypotheses to be explored in this thesis are therefore the following:

- **Hyphothesis 1**: Filtering the corpus to actively downsample and thus reduce the amount of tweets from highly active users to a defined limit has, depending on the chosen limit, a positive effect on the results of the downstream tasks, as the learned representations of the words reflect the global word meaning.

And the other, secondary hypotheses:

- **Hyphothesis 2**: The length of a tweet positively correlates with its linguistic information (Tan et al., 2015; Boot et al., 2019; Gligorić et al., 2018). Selecting only tweets above a certain minimum length has a positive influence on the results of the downstream tasks.
- **Hyphothesis 3**: Selecting a subset of different years or periods of content does not change the linguistic quality of the tweets. Thus, models trained on different years should, assuming a similar token count, perform similarly.

There is an underlying contrast between the expected effectiveness of the filter choices and the resulting data amount. A general assumption here would be that

the number of data, in the form of tokens, is decisive for the performance of the models, regardless of the filtering decision. To evaluate this, I propose combinations of the filters above which aim to evaluate how exactly filtering decisions affect the performances of the models and to which extent input data has an influence.

1.2 Structure of this Thesis

The remainder of this thesis is divided into the following sections:

Section 2 - **Background**: This section delves into the fundamental concepts that underpin the experiments, providing essential context for understanding the subsequent sections. Also, relevant prior research and literature related to the topic are presented, highlighting existing work and providing motivation on why this thesis has scientific merit.
Section 3 - **Data**: Here, the corpus used in this thesis and its creation process and properties are presented. Furthermore, the evaluation sets are also briefly introduced.
Section 4 - **Method**: This section outlines the methodology, meaning the experimental setup for this thesis, which is used to address the hypotheses. Also, the various preprocessing steps, filters, and the resulting models employed in this thesis are described, providing insight into the training and evaluation methods.
Section 5 - **Results**: This section presents the findings and outcomes of this thesis, with data analysis and interpretation for both downstream tasks.
Section 6 - **Conclusion and Future Work**: The summary section provides a concise recap of the thesis' key points, emphasizing the most significant insights and contributions while also providing possible extensions.
Bibliography: This section lists all the references cited throughout the thesis.
Section 7 - **Appendix**: The appendix contains additional supplementary material in the form of tables containing all results of all models for both downstream tasks.

2 Background

This section focuses on introducing and presenting the necessary basic background knowledge that gives an overview of the methods and techniques of the research area used in this thesis, while also providing an introduction to the history of word embeddings.

2.1 Vector Representations of Words

The history of word embeddings can be traced back to the beginnings of distributional semantics. The theory of distributional semantics is based on the concept that information about the context of a word already contains significant and supportable linguistic information, i.e. words that occur in the same context and vicinity tend to have similar meanings (Harris, 1954). The underlying idea of this so-called distributional hypothesis was popularized by Firth who philosophized in the 1950s that "a word is characterized by the company it keeps" (Firth, 1957). The earliest attempts of deriving or constructing features that share semantic similarities were made by Osgood (1964). It took until the 1990s for the first methods for the use of automatically generated contextual properties to emerge in the form of *Latent Semantic Analysis (LSA)* (Deerwester et al., 1990) or *Simple Recurrent Networks* (Elman, 1990) amongst others.

Regardless of the selected model architecture, a common trend nowadays is using a vector space, which provides a spatial representation of the word's meaning (Wendlandt et al., 2018). This is based on the fact that vector similarity is the only tangible information present in word-space models, meaning that semantically related words are close while unrelated words are distant (Schütze, 1993). However, there are many different ways to move from the assumed distributional statistic to the respective geometric representation of a low-dimensional word-vector space. These include context vectors, probabilistic approaches, and co-occurrence matrices amongst others (Sahlgren, 2006).

At a basic level, a word vector is just a vector of weights. Using a simple encoding method such as one-hot encoding, each element in the vector corresponds to exactly one word in the vocabulary. The encoding of a given word is simply the vector in which the corresponding element is set to one, and all other elements are zero. However, if such an encoding is used, no meaningful mathematical operation can be performed apart from equality testing. As a result, demand for more elaborate embedding methods arose.

One of the most popular methods for generating word embeddings is using the word2vec model, introduced by Mikolov et al. (2013a). Word2Vec operates on the same principle that words appearing in similar contexts are likely to have similar meanings. It can create dense, low-dimensional word representations that capture semantic relationships by using a distributed representation of each word by representing a word by a vector with hundreds of dimensions. (Goldberg and Levy, 2014) This is mainly a matter of converting words into floating-point numbers. In this respect, vectors with several dimensions are created, where each dimension contains a certain characteristic of information. Each vector forms a data point in an n-dimensional space. Consequently, the vectors reflect the structure of the entire corpus. The following section will describe the word2vec model in more detail.

2.2 Word2Vec

Mikolov et al. (2013a) introduced the **word2vec model**, which is an efficient method for learning vector representations of words from large amounts of unstructured text data. On a surface level, Word2Vec consists of a shallow, two-level neural network. It maps each word to a fixed-length vector, and these vectors can better express the similarity and analogy relationship among words. A further distinction is made between two different architectures: (i) the *Continuous Bag-of-Words Model* and (ii) the *Skip-gram Model*. Later, an extension of the original Skip-gram architecture was provided in Mikolov et al. (2013b). The main difference is that the CBOW architecture predicts the current word based on the context, and the Skip-gram architecture predicts surrounding words given the current word (Mikolov et al., 2013a). A graphical representation of the architecture of the models can be found in figure 1. As this thesis focuses on the Skip-gram architecture, there will be an in-depth look into its structural design and its mode of operation.

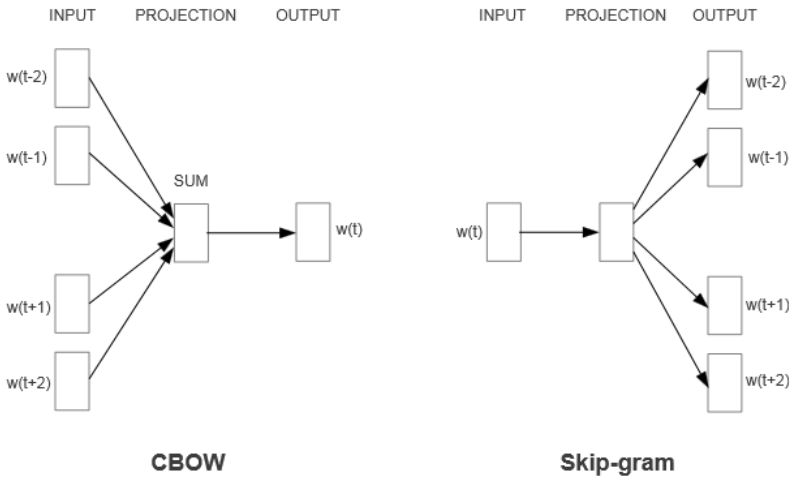

Figure 1: The architectures of the models. The CBOW architecture predicts the current word based on the context and the Skip-gram predicts surrounding words given the current word. Mikolov et al. (2013a)

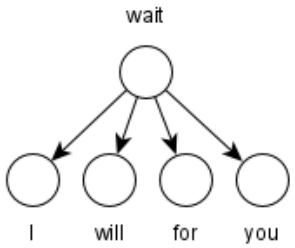

Figure 2: An example of the sequence *I will wait for you* in the Skip-gram architecture. The model considers the conditional probability of the surrounding words given a word.

Figure 2 shows the sentence *"I will wait for you"*. Assuming *wait* is the central word with a context window of 2, the Skip-gram model now considers the conditional probability for generating the context words (2 to the left, and 2 to the right) of the words *I, will, for, you* by assuming that the words in its context are independently generated given an input word in the center:

$$P(I, will, for, you|wait) = P(I|wait) \cdot P(will|wait) \cdot P(for|wait) \cdot P(you|will)$$

2.2.1 Skip-Gram

As previously stated, the Skip-gram model assumes that for each word in a sequence of text, you can use this word to generate its surrounding words. Each word w has two $(d, 1)-$dimensional vector representations to calculate its conditional probabilities with $|V|$ being the total number of words and i being its index in the dictionary: the center word vector $v_i \in R^d$, and the context word vector $u_i \in R^d$. Generating a context word w_o (with o being its index in the dictionary) is now described by the conditional probability given a center word w_c (with c being its index in the dictionary) and applying a softmax operation of the vector dot products u and v:

$$P(w_o|w_c) = \frac{exp(u_o^T v_c)}{\sum_i^V exp(u_i^T v_c)} \tag{1}$$

Assuming a text sequence of length T $(w_1, w_2, w_3, \ldots, w_T)$ with the word at each time step t is noted as w_t, and that context words are independently generated given any center word and context window size c (not the same c as the center word), the likelihood function of this model is described as the probability of generating all context words given any center word as input:

$$L(u_i, v_i) = \prod_{t=1}^{T} \prod_{-c \leq j \leq +c, j \neq 0} P(w_{t+j}|w_t) \tag{2}$$

As the parameters of the Skip-gram model are the center and context word vectors for each word in the vocabulary, the objective of the Skip-gram model is then to minimize the negative log-likelihood which in return maximizes the likelihood function:

$$J(u_i, v_i) = -\frac{1}{T} log(L(u_i, v_i)) = \frac{1}{T} \sum_{t=1}^{T} \sum_{-c \leq j \leq +c, j \neq 0} logP(w_{t+j}|w_t) \tag{3}$$

According to Mikolov et al. (2013a) larger values for c lead to better results but are computationally harder and thus take more time. This function can be solved by using stochastic gradient descent to minimize the loss by using the gradients of the log conditional probability with the center word vector and the context word vector. This can be written as:

$$logP(w_o|w_c) = u_o^T v_c - log(\sum_{i}^{V} exp(u_i^T v_c)) \tag{4}$$

In Mikolov et al. (2013b) they expanded on the original computationally expensive softmax function by introducing two new concepts, hierarchical softmax, and negative sampling. Hierarchical softmax uses a tree representation of the output layer and therefore approximates the full softmax by only evaluating a subset of its original nodes which reduces the total amount of calculations. An alternatively proposed solution is negative sampling which reduces computation by sampling just N negative instances along with the central word instead of sampling the whole vocabulary. The selection of words to be sampled depends on their frequency. Essentially, the probability of selecting a word as a negative sample is related to its frequency, with more frequent words being more likely to be selected as negative samples. Each word

is assigned a weight equal to its frequency (word count) raised to the 3/4 power. The likelihood of picking a word is its weight divided by the total of all weights (Mikolov et al., 2013b). A more detailed explanation can be found in Goldberg and Levy (2014).

2.3 Downstream Tasks

The quality of the word embeddings can be measured by applying them to standard downstream tasks. The results of the tasks give an indication of how well the model has understood the semantic meanings of the words of the corpus and in which context they are employed. One of the most common methodologies to assess the quality of word embeddings is to assess them with specific test sets aimed to evaluate the models' performance on different tasks. In this thesis, two downstream tasks, namely word similarity and lexical semantic change were selected to evaluate the learned models.

In particular, the concept of word similarity plays a common role in confirming the quality of word embeddings as this is a fairly general and robust task (Doval et al., 2020), whereas lexical semantic change is a more specific and niche task (Hamilton et al., 2016), which nevertheless has its importance, as the word meanings play an immensely important role and one can easily see from the results to what extent a change in word meaning has taken place by comparing the word vectors.

Essentially, the previous sections also highlighted the problem of very active users in user-generated corpora for semantic analyses based on word embeddings. Naturally, for very frequently used words, no significant changes in frequencies are expected by the above-mentioned users. However, it may well be that a few users use a certain word in different contexts than usual and the model learns these skewed representations of word use since it has no basis for comparison which may alter the resulting word vectors in an unexpected way. It can even go so far that a single user, through the strikingly frequent use of a single word, significantly influences and changes

the resulting word vector. This could lead to the word being misrepresented and significantly alter the generality of the word's meaning, which is a problem for all subsequent tasks. The two tasks selected for this thesis will briefly be introduced in the following sections.

2.3.1 Word Similarity

Word Similarity, in broad terms, is defined as the degree of likeness between two words in terms of their meaning or contextual content. It can be viewed as a measurement of how closely related two words are in the definition and usage (Navigli and Martelli, 2019; Elekes et al., 2017). There are many different methods and metrics to quantify the relation between two words (Elekes et al., 2017; Navigli and Martelli, 2019) with a general distinction into knowledge-based approaches and distributional approaches. Navigli and Martelli (2019) define the first approach by its mode of operation of exploiting explicit representations of meaning derived from wide-coverage lexical-semantic knowledge resources and the second approach by its formal distributional semantics basis, which aims to exploit the statistical distribution of words within unstructured text. In this thesis, I will use the second approach, as I have an unstructured corpus from which I'll derive the needed word similarities by computing word vectors.

The measure to compare the degree of semantic similarity is done by comparing two (or more) resulting vector representations. Widely used amongst both approaches is the measurement of semantic similarity via cosine similarity (Artetxe et al., 2018; Navigli and Martelli, 2019). Mathematically, the task is defined as two words a and b with their corresponding word vectors a_1 and b_1 in a n-dimensional vector space $a, b \in R^n$. The similarity between the word vectors can then be obtained by computing the cosine similarity between the vectors of the word pair:

$$cosine\_sim(a_1, b_1) = \frac{a_1 \cdot b_1}{||a_1||||b_1||}$$

In intrinsic word similarity evaluation, word pairs along with their similarity rating as judged by human annotators are provided. The task is then defined as the mea-

surement of the distance between the calculated distance measure and the average over the human annotators (Artetxe et al., 2018). The similarity scores are computed for all lists of word pairs in the test set and then sorted according to their computed vector-space similarity and human similarity. The more similar they are, the better are the embeddings. Computing Spearman's correlation (Myers et al., 2010) or Pearson's correlation (Freedman et al., 2007) between these ranked lists results in a score that reflects how well the learned word vectors capture the concept of similarity.

The evaluation of word embeddings on a dataset makes it easy to compare one of the models created for a task to other models created for a task. This and the fact that the evaluation is computationally fast and easy allows us to easily understand which parameters (or in this case, filters) have a larger impact on the score. Section 4.6.1 will further elaborate on the evaluation method and sets used for this thesis.

2.3.2 Semantic Change

(Lexical) semantic change detection, in broad terms, is defined as the detection of word meaning change. (Tahmasebi et al., 2021) There are multiple explanations for the shift of word meaning, which include cultural, technological, and linguistic factors (Hamilton et al., 2016). The ever-improving neural models for creating word embeddings also led to an increased interest in this field. (Hamilton et al., 2016; Kutuzov et al., 2018; Schlechtweg et al., 2019). A further distinction is made between synchronic and diachronic semantic change detection (SCD) (Tahmasebi et al., 2021).

Diachronic SCD evaluates shifts in meaning over time by measuring the change of word embeddings trained on corpora based on different time periods. Synchronic SCD also evaluates shifts in meaning but doesn't necessarily base the assumption of change on time alone, but rather on other focus areas like domain (Schlechtweg et al., 2019), where a difference in word meaning compared to the general usage is to be expected.

In this thesis, the semantic shift of words, as contact-induced synchronic change, will be investigated by evaluating the different word usage and meanings of Canadian English words across different regions, where one region is affected by the contact-induced influence of French (Quebec English) and the other is not. It has to be mentioned, that due to cultural and geopolitical reasons, an underlying diachronic explanation for eventual changes cannot be excluded. Section 3, which focuses on the corpus, will further elaborate on the creation process of the corpus.

Computational approaches to semantic change detection in the last years have largely used vector space models which dominate the current research (Kaiser et al., 2021). There are many different methods with different alignments, and similarity measures to fulfill this task Schlechtweg et al. (2019). Commonly used are SGNS models with the *Orthogonal Procrustes* alignment and the cosine distance/similarity to measure the results (Schlechtweg et al., 2019; Kaiser et al., 2020). The selection of the orthogonal procrust alignment method leads to a separate training of the SGNS models for each corpus resulting in two different word matrices which have to be aligned according to the method proposed by Hamilton et al. (2016). The measurement and the subsequent evaluation of the results between the two words from the two now-learned SGNS-based representations is similarly done as described in the previous section.

In such a representation, and for the purpose of this thesis, for each word w_i a vector representation w_i^c for each of the three different subcorpora of the corpus $w_i^{Montreal}, w_i^{Toronto}, w_i^{Vancouver}$ is computed. The distance between the respective word vectors for the same word is then computed similarly to the previous section, where a distance metric, like the cosine distance, is used to compute the rate at which the words have different meanings in the regional areas. Section 4.6.2 will further elaborate on the evaluation process of this thesis by introducing the evaluation method and test set.

2.4 Related Work

The methodological foundation of this thesis relies on word vectors created by word embeddings. Word embeddings are a practical implementation of the concept of distributional semantics as shown in the previous sections by creating dense, low-dimensional vector representations of words learned from corpora.

In current research, a distinction is made between two different types of distributional semantic models, there are (i) (static) type-based models and (ii) (dynamic and contextual) token-based models (Tahmasebi et al., 2021). The crucial difference between the two approaches lies in the distinction between models building one representation of the word's meaning for each word (token-based) and models building a representation of the word's meaning by aggregating over the word's uses (type-based).

While type-based models such as word2vec's Skip-Gram with Negative Sampling model (SGNS) (Mikolov et al., 2013a;b) or GloVe (Pennington et al., 2014) are prone to the aforementioned limitations of the architecture, the results of SGNS, in particular, are outperforming all token-based models for lexical semantic change detection on many datasets (Laicher et al., 2020; Kaiser et al., 2020). Due to its fast and easy implementation, allowing for multiple different combinations of parameters and alignment types, SGNS is a widely used resource for lexical semantic change detection (Shoemark et al., 2019; Tahmasebi et al., 2021; Kaiser et al., 2020) and therefore the embeddings used in this thesis are SGNS-based. Generally, word2vec models are also considered to be the most popular and yet one of the most successful models for the task of word similarity (Navigli and Martelli, 2019; Wang et al., 2019).

There has been a substantial amount of research done using word2vec models for both downstream tasks for this thesis (Tahmasebi et al., 2021; Shoemark et al., 2019; Elekes et al., 2017; Schlechtweg et al., 2020; Kaiser et al., 2020). In general, word2vec methods have been applied to many different tasks trained on different

types of data sets (Naseem et al., 2020; Babić et al., 2020; Tan et al., 2015). Examining how different types and approaches to create and apply word2vec models respond to different circumstances, given the wide range of parameters, input data, alignment methods, and downstream tasks is too extensive to provide an overview here.

The analysis of properties and characteristics of Twitter and corpora derived from Twitter data has gained some traction in academia with González (2015) offering a statistical analysis of Twitter corpora and their properties and the difference between formal and informal tweets by the usage of Twitter-specific functions. Neubig and Duh (2013) investigate the information gain per character on Twitter for many different languages and come to the conclusion that small differences in size lead to more information gain. Şahinuç and Toraman (2021) investigate the impact of the length of a tweet with similar findings. Likewise, Gligorić et al. (2018); Boot et al. (2019) evaluate if the alteration of the constraints (140 to 280 characters) of maximally allowed characters on tweets leads to more linguistic information.

Such findings aren't prone to be domain and task specific as (Klein et al., 2021; Taie et al., 2019; Bian et al., 2008; Sastry, 2012) amongst others use Twitter corpora for further tasks and offer similar descriptions as above about the statistical properties of their corpora, which ultimately is summarized by Moreno-Ortiz and García-Gámez (2023) by evaluating the methods of dealing with large corpora based on social media data.

The investigation of word2vec models applied to Twitter corpora and how the results differ in comparison to other sources has received some attention as Li et al. (2017) investigate how word2vec models trained on different Twitter corpora by omitting certain preprocessing steps like spam detection and removal behave and come to the conclusion that the more context in the form of longer tweets the better the models perform. Similarly, Tan et al. (2015) compare word2vec models trained on different lexical corpora, Twitter and Wikipedia, and arrive at the result that there

are certain characteristic words that differentiate the two. For example, they find that Twitter corpora contain much different word vectors for words that are used colloquially and in abbreviations. Eventually, Doval et al. (2020) offer a large-scale analysis of different word embedding models trained on noisy Twitter data and also word similarity to evaluate their method of using bridge words to fill out gaps of content.

However, there is not too much research on the challenges of uneven data distribution using Twitter data and word2vec models. Many recognize the inequality of data distribution in user-generated corpora but limit themselves to setting common boundaries above which users' tweets are subsampled or simply subsample tweets in general (Miletic et al., 2021; Shoemark et al., 2019; Tan et al., 2015; Moreno-Ortiz and García-Gámez, 2023) but don't systematically investigate the effectiveness of such filtering techniques. It has been known that word2vec models are prone to instabilities (Antoniak and Mimno, 2018) especially when they are trained on smaller corpora (Wendlandt et al., 2018).

As such, based on the corpus created by Miletic et al. (2020) I will investigate how filtering methods with the goal to select a better distribution of data across users and the resulting corpora affect the results of word similarity and lexical semantic change detection by training word2vec models and comparing their results.

3 Data

The objective of this section is to (i) present the used corpus and (ii) the used evaluation sets. Section 3.1 will present and introduce the corpus used for this thesis, and the subsequent sections will introduce and provide an overview of the used evaluation sets for both tasks of this thesis.

3.1 Corpus

Miletic et al. (2020) introduce and present the **CanEn**[3] corpus for this thesis and thus the foundation all work is based on. As previously discussed in section 2, the corpus was created to study regional variation in Canadian English and, more specifically, to investigate if any contact-induced semantic shifts occurred in Quebec English. For this, three different subcorpora, all related to three large cities in Canada, one of which where a semantic shift due to contact is expected and two of which where this shift isn't expected to serve as a further sanity check, have been defined by differentiating between the geological location of each user. Namely, these are the Montreal, Toronto, and Vancouver corpora. Each of which will be subsequently treated as its own separate corpus, despite all of them being part of a complete corpus, to avoid misunderstandings due to the wording, if necessary. Section 3.1.1 will briefly summarize the creation process of the corpus while section 3.1.2 will provide a description of the corpus and its properties with a special focus on the distribution of content across users.

3.1.1 Creation of the Corpus

Miletic et al. (2020) define the data collection process by creating a pipeline consisting of two steps: (i) a data collection step that seeks to identify users in the geographic areas of interest, and (ii) a subsequent crawl of their timelines. They further elaborate on their choice of geographic areas and the initial tweet collection

[3]http://redac.univ-tlse2.fr/corpora/canen.html

by providing sociolinguistic justifications for their choices. Furthermore, the derived tweets were filtered by location, language, and near-duplicate detection to ensure that the retrieved data is not only appropriate but also usable for further tasks in the field of natural language processing. Figure 3 provides an overview of the data collection and subsequent filtering pipeline of the CanEn corpus.

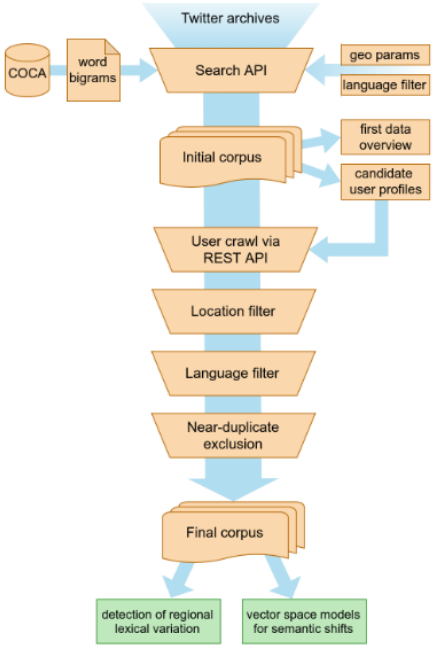

Figure 3: Data collection pipeline detailing all major steps for the creation of the CanEn corpus (Miletic et al., 2020).

3.1.2 Description of the Corpus

The complete corpus contains 78.8 million tweets posted by 196431 individuals. After tokenizing the corpus with an implementation of nltk's *TreebankWordTokenizer* (Bird et al., 2009) this leaves us with 1.2 billion tokens for the corpus. Table 1 shows a more detailed view of the presented properties. There are, on average, 325 tweets per user for the Montreal subcorpus, 443 tweets per user for the Toronto subcorpus,

and 449 tweets per user for the Vancouver subcorpus, which results in a global average of 401 tweets per user. The mean amount of tokens per tweet is, on average, 14.8 for the Montreal corpus, 15.8 for the Toronto corpus, and 16.2 for the Vancouver corpus, which results in a global average mean token amount per tweet, equivalent to the length of a tweet, of 15.4.

Corpus	Users	Tweets	Tokens
Montreal	72 305	23 469 526	352 202 123
Toronto	64 163	28 442 928	437 301 043
Vancouver	59 692	26 924 158	428 998 623
Complete	196 431	78 836 612	1 218 501 789

Table 1: Structure of the corpora, indicating the number of users, tweets, and tokens per corpus

3.1.2.1 Distribution of Tweets across Users

The distribution of tweets can be seen in figure 4 depicting histograms for each corpus. The distribution follows a similar trend seen in user-generated corpora where the number of users with only a small number of tweets is much larger than the number of users with a lot of tweets (Sastry, 2012). The number of users decreases significantly as the number of tweets increases so that already the number of users with hundreds of tweets is only a fraction of the users with less than 60 tweets. This sort of distribution is very common amongst user-generated corpora and thus the cumulative distribution function shown in 4 shows that the large number of users with less than 60 tweets already accounts for roughly half of all tweets in the entire Montreal corpus. Another interesting find is the fact that the most active 1% of users account for more than 6.2% of all tweets. Similar trends can be observed in the two other corpora. Considering that a non-negligible percentage of the corpus is constructed by these highly-active users, a further investigation into how their tweets affect the resulting word embeddings is warranted.

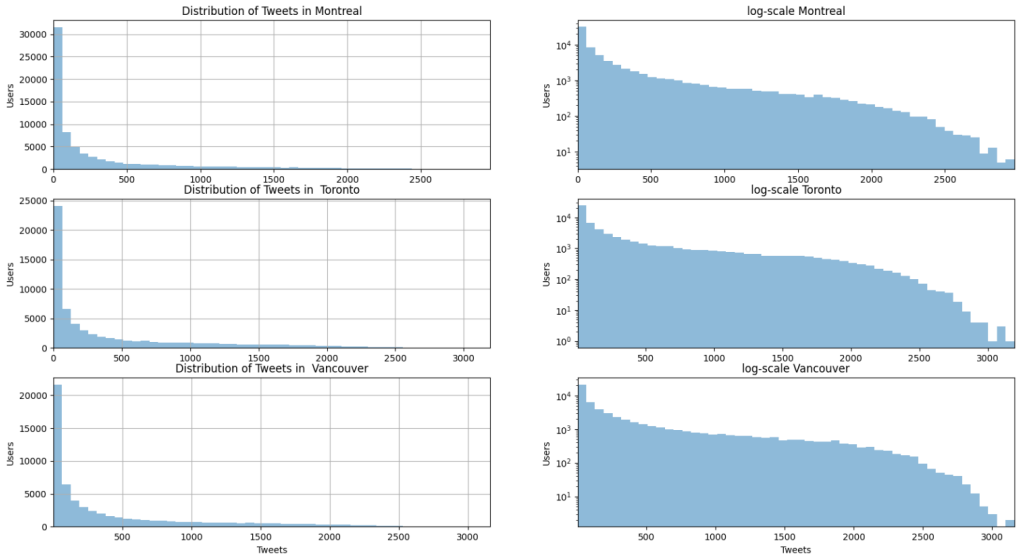

Figure 4: Histograms of the distribution of tweets per user across the three corpora using (i) a normal scale on the left side and (ii) a logarithmic scale on the right side.

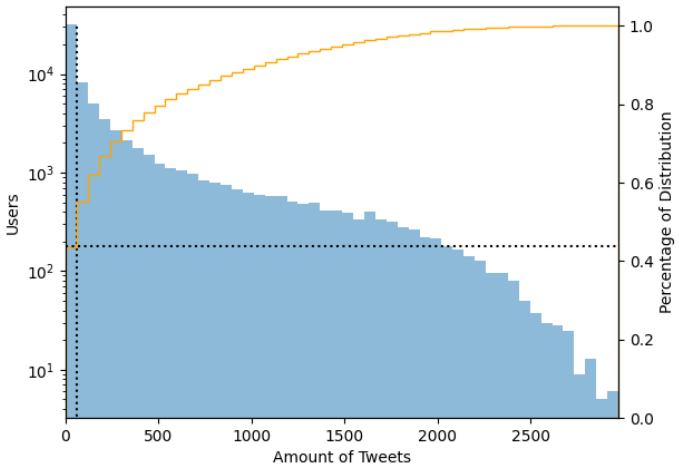

Figure 5: Histogram of the distribution of tweets oer user of the Montreal corpus including a cumulative distribution function shown in orange.

3.2 Test Sets

This section presents the evaluation sets of both, word similarity and semantic change detection.

3.2.1 Semantic Shift

Miletic et al. (2021) introduce and present the test set (CanEn test set[4]) for this thesis. As mentioned in the previous sections, it was not possible to use standard resources to investigate if any words underwent contact-induced semantic change. Thus, a new evaluation set based on this corpus had to be created which I'll briefly present in the following segment.

3.2.1.1 CanEn Test Set

As stated in the paper, this evaluation set was specifically created for this particular corpus as it seeks to find words that underwent contact-induced semantic shifts in Quebec English. To achieve this, Miletic et al. (2021) define the task to detect lexical semantic changes similar to other recent works proposed by Schlechtweg et al. (2020) as a binary clarification problem.

Therefore, the words of this 80-word test set were divided into two groups: One where a contact-induced semantic change was to be expected according to sociolinguistic literature (*class 1*), and the other where no change was to be expected which served as a control group (*class 0*). Each line in the file contains a word, its respective POS tag and its semantic change class. Table 2 shows an example of the set.

[4]http://redac.univ-tlse2.fr/misc/canenTestset.html

word	POS	class
chalet	N	1
formidable	A	1
ought	V	0
hunch	N	0

Table 2: Structure of the CanEn test set using exemplary words.

3.2.2 Word Similarity

There are several, different data sets for word similarity. These differ in approach and application so that a wide range of challenges in word similarity (relatedness, association, similarity) can be queried but their design is generally similar according to Faruqui et al. (2016). Namely, these evaluation sets include a tabular structure in which each row has exactly one-word pair and all associated metrics in the form of columns. Generally, two columns represent the respective word pairs and the remaining columns represent metrics such as the average score of all human annotators. MEN, WordSim-353 and SimLex-999 are standard resources to evaluate models on word similarity and relatedness and were therefore chosen as the test sets for this thesis. (Doval et al., 2020) Each of these I'll briefly describe in the following segments.

3.2.2.1 SimLex-999

SimLex-999 (Hill et al., 2015) is a resource for evaluating models that learn the meaning of words. It focuses on similarity rather than other similar concepts so that pairs of words that are associated but not actually similar (*Djokovic, Tennis)* have a low rating and thus allows general-purpose evaluations of semantic models. It contains 999 word pairs with a scale from 0 to 10. A distinction from the next test set (*WordSim-353*) can be found in Table 3 showing that the word pair *clothes - closet* receives a low score even though they are conceptually related.

Word Pair	SimLex-999	WordSim-353
clothes - closet	1.96	8.00

Table 3: Comparison of the scores given by SimLex-999 and WordSim-353 for two selected word pairs

3.2.2.2 WordSim353

WordSim353 (Finkelstein et al., 2001) is a human-constructed test set for measuring word relatedness and similarity. It contains a set of English word pairs along with their human-assigned scores. Agirre et al. (2009) propose a split of the original test set into two different subsets, one for evaluating similarity *WordSim353-Similarity*, and the other for evaluating relatedness *WordSim353-Relatedness)*, which I also used to compare the resulting models. Following the split, the similarity gold standard set contains 203 lines of word pairs while the relatedness gold standard set contains 252 lines of word pairs. The values for the scores range from close to 0 (*no similarity or relatedness between the two words*) to 10 (*identical word*). The WordSim-353-REL set contains no pairs of similar concepts while the WordSim-353-SIM contains similar or unassociated pairs.

3.2.2.3 MEN

MEN (acronym for Marco-Elia-Nam) (Bruni et al., 2014) is a human-constructed test set for measuring word relatedness. It contains 3000 word pairs assessed by their semantic relatedness on a scale from 0 to 50. The structure is similar to the evaluation sets described above. Due to the large number of word pairs, the test set is particularly interesting for its wide scope of words, especially since for some models it may occur that words in the alternative test sets do not necessarily appear in the vocabulary of the corpus.

4 Method

This section describes the method and experimental setup of this thesis. A subsection is dedicated to each individual step in order to describe each process as precisely as possible. Thus, section 4.1 provides an overview of the methodology of the thesis, section 4.2 presents the individual text preprocessing steps of the original corpus, section 4.3 describes and explains the chosen filtering criteria, section 4.4 describes the general training process including the algorithm choice, its parameters and configurations of the hardware, and section 4.5 presents properties the resulting models. Finally, section 4.6 provides a review of the evaluation process for the models.

4.1 Overview of the Experimental Setup

The setup for the experiment can be loosely divided into four different phases. First, the text of the corpus is preprocessed. Then, the corpus is modified by already defined filters that use certain properties of Twitter to select specifically selected tweets into their own, smaller corpora. The general filter categories are the length of the tweet, the number of tweets per user, and the year the tweet was written. There are furthermore, combinations of filter categories, which result in additional corpora. Each of these corpora goes through the further steps separately. Subsequently, word embeddings are created from all corpora according to the same training parameters and features using SGNS, which are then analyzed and evaluated based on the two downstream tasks; word similarity and semantic change detection. The average of the results of the four different test sets for word similarity is used for comparison. SCD uses only one test set, created specifically for this task. The aim of this schematic design is to provide a uniform basis of comparison for the effectiveness of filtering decisions based on Twitter corpora-specific distributions of tweets. Figure 6 shows the experimental setup for this thesis. The following section will go into more detail for each respective step of the setup.

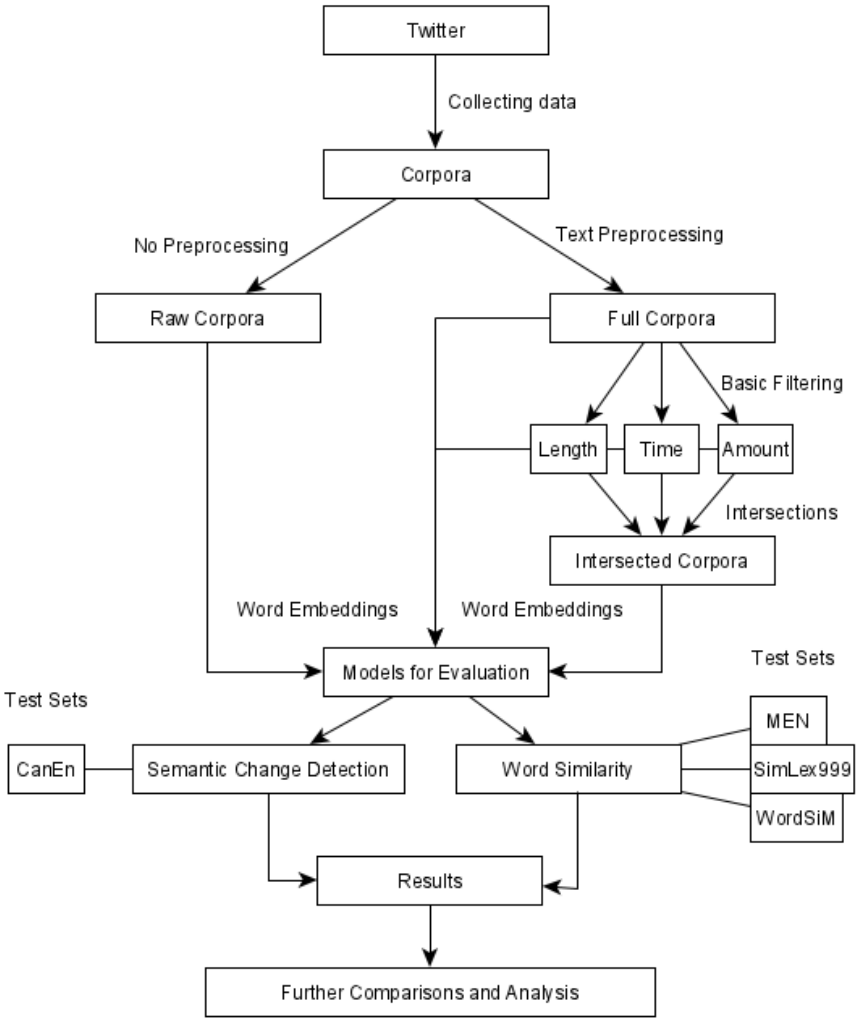

Figure 6: Experimental setup of this thesis which includes the four main steps: (i) text preprocessing, (ii) filtering of the data, and the subsequent (iii) training and (iv) evaluation of the learned word embeddings on the basis of two selected downstream tasks; semantic change detection and word similarity and their respective test sets.

4.2 Text Preprocessing

The preprocessing and cleaning of data in the corpus is an essential part of setting up an experiment in the field of natural language processing. This is especially important for user-generated data such as corpora of tweets (Chai, 2023). In contrast to corpora that are created according to predefined criteria, such as corpora that are based on the usage of sentences from journals, and newspaper articles (Oberbichler and Pfanzelter, 2021) or created by annotators relying on services like Mechanical Turk (El-Haj et al., 2010), Twitter tweets, similar to data from other social media platforms, are inherently user-generated comments and therefore their content is also freely left to the respective user and thus unstructured (Taie et al., 2019).

The fact that, apart from the fixed maximal possible length of a tweet, users have no restrictions on what they write, especially in terms of the type, the content, and the spelling, makes Twitter data inherently noisy. Due to the fact that (almost) everything is allowed, and also comparatively much can be written, there are clear differences between the qualities of the tweets. Not necessarily limited to Twitter, Bian et al. (2008) as early as 2008, found that data quality in such environments ranges from linguistically valuable to spam and malicious content. This trend, which has been continuously observed over the last decade as shown by Tan et al. (2015); Klein et al. (2021), justifies meticulous and large-scale preprocessing of Twitter corpora as absolutely necessary.

The process of text preprocessing has received a lot of attention in current research with Chai (2023) offering a meticulously detailed survey on the effectiveness of the most common steps such as the removal of stop words or cleaning of misspelled words. Bao et al. (2014); Palomino and Aider (2022); Ramachandran and Parvathi (2019); Symeonidis et al. (2018) amongst others also provide an investigation and an analysis of the standard preprocessing techniques in the field of natural language processing, with a special focus on sentiment analysis.

Miletic et al. (2020) have already preprocessed the corpus in the form of detecting and eliminating spam and other malicious content, detecting and removing near-duplicates, and other common preprocessing steps such as language identification. The content of the tweets themselves, have not been altered. Consequently, when creating the text preprocessing pipeline, I followed common steps for cleaning and preprocessing Twitter data, which are listed and briefly explained below:

1. *Removing Emojis*: Removing any emojis present in the corpus. Emojis are pictorial representations of emotions, ideas, or things amongst others.
2. *Removing Emoticons*: Removing emoticons, images made up of symbols like ":D", entirely.
3. *Lowercase*: Converting all text to lowercase.
4. *Replacing Mentions*: Replacing user mentions (for example @milovamn) with a generic mention (@username).
5. *Removing Links*: Removing any URLs, hyperlinks or links present in the text.
6. *Removing Hashtags*: Removing hashtags (words or tokens preceded by '#') present in the text.
7. *Removing Numbers*: Removing any numerical digits from the text.
8. *Splitting Tweets into Sentences*: Dividing each tweet into individual sentences.
9. *Splitting Sentences into Tokens*: Splitting each sentence into its individual tokens.
10. *Removing Punctuation*: Removing any punctuation marks from the tokens.
11. *Lemmatization*: Converting each token to its root form (lemma), which reduces inflected words to their base form and normalizes the text for further analysis.

12. *Removing stop words*: In this step, common words that do not carry significant meaning and are often considered noise in natural language processing tasks are eliminated from the text. These words, known as stop words, include frequently occurring words such as ”the” or ”is”. By removing these words, the text is streamlined, and the focus is shifted to more relevant and informative content. This is important as the sampling process of Skip-gram gives the most frequent n words a lower weight. Removing stop words entirely, alters the words that usually get a lower weight Antoniak and Mimno (2018).

Figure 7 shows a graphical overview of all steps and their order in this preprocessing pipeline. It i

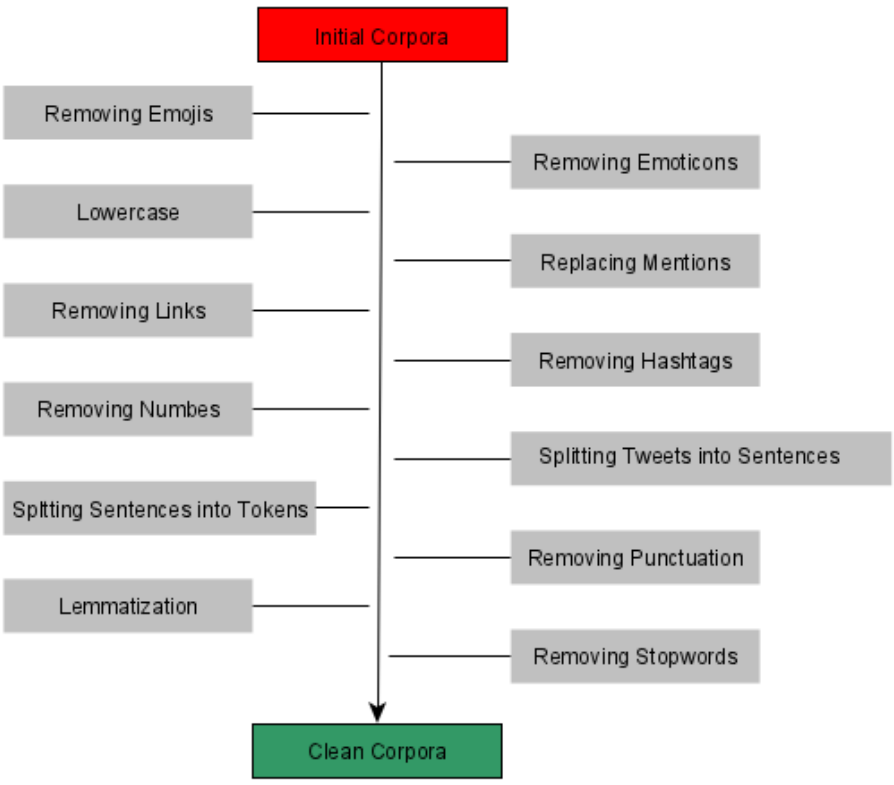

Figure 7: All steps taken during text preprocessing of the corpus.

4.3 Filters

The initial filtering of the original corpus can be divided into three different filtering categories. In the first category, filtering is based on the original date of the tweets. The second category filters by the length of the tweets and the third category filters by the number of tweets per user. Thus, the categories can be roughly divided into two classes. On the one hand, the content of the tweet itself is considered, and on the other hand, meta-information about the tweet is taken into account. In order to guarantee a generally applicable analysis of the influence of filter processes, the individual details of the respective filters must be set in such a way that the entire process can be reproduced and monitored. For this purpose, separate variants were defined for each individual category of filters, which are easily distinguishable from each other. Thus, several properties and their variants of the tweets can be analyzed. This is especially important in the second filter category, where the filtering was deliberately set to certain conditions that allow comparisons even across the most minuscule of differences.

4.3.1 No Filters

Two versions of the original corpus were used as a starting point for the models to allow for comparison against the models based on filtered corpora. Namely, the raw, i.e. not preprocessed corpus, and the full, i.e. the completely preprocessed but not filtered corpus. Both of these serve as baselines for any comparisons and the conclusions that arise from the comparisons.

4.3.2 Basic Filters

This section will introduce all filter categories and their traits and offer insight into the resulting properties. It also serves as a foundation to justify further decisions based on the results of the filtering process.

Filter Category 1 - Time: A distinction between active periods of time per user has been proposed in Miletic et al. (2021) where the cut-off was chosen to be 2016 to not have any synchronic effects. On a more general level, these two variants of this filtering category were used because I wanted to investigate the differences in properties between earlier and later tweets. It should be clear that combinations with the filter of the first five years cannot provide meaningful results due to the low number of tweets and the associated low number of tokens. Table 4 shows the number of tweets for each period.

- First 5 Years: Resulting corpus only retains tweets that were written in the first 5 years of the corpus.
- Last 5 Years: Resulting corpus only retains tweets that were written in the last 5 years of the corpus.

Year	Montreal	Toronto	Vancouver
2006	32	21	9
2007	3 007	1 976	2 145
2008	26 129	21 896	29 361
2009	231 312	281 623	349 367
2010	406 465	531 518	651 818
2015	2 395 221	2 948 874	2 719 550
2016	2 686 443	3 185 845	2 851 135
2017	3 166 136	3 504 498	3 297 711
2018	4 445 400	4 802 701	4 509 463
2019	2 821 589	3 157 332	2 835 450

Table 4: Comparison of the number of tweets available for each corpus across the first and last five years. It can easily be seen that the number of tweets per year is increasing significantly.

Filter Category 2 - Length: It is genuinely accepted that there is a correlation between tweet length and data quality (Boot et al., 2019; Gligorić et al., 2018; Şahinuç and Toraman, 2021). Due to many works in research proposing a minimum tweet length as seen in Miletic et al. (2021); Moreno-Ortiz and García-Gámez (2023); Neubig and Duh (2013) each tweet got to have to be used for further analysis, the distinction between different length cutoffs is worth of an in-depth investigation. Thus, I've defined four different approaches to evaluate reasonable cutoffs with table 5 showing the actual results cutoff values:

- 25%: Resulting corpus only contains tweets whose length is above the lower quartile.
- 50%: Resulting corpus only contains tweets whose length is above the overall median length.
- 75%: Resulting corpus only contains tweets whose length is above the upper quartile.
- mean: Resulting corpus only contains tweets whose length is above the overall mean length.

Furthermore, I've defined two different variants which aim to analyze both extremes of the length distribution, namely the *min* which aims to capture tweets below the minimum length + d and the *man* filter which aims to capture tweets above the maximum length - d per user. The value $d = 10$ was chosen as it yielded an appropriate sample size.

- min: Resulting corpus only contains tweets that are smaller than the minimum average length per user + 10.
- max: Resulting corpus only contains tweets that are larger than the maximum average length per user - 10.

The distinctions in table 5 are small, but according to Neubig and Duh (2013) who calculated the amount of information included in one character in social media corpora for multiple languages and came to the result that small differences in character size lead to large information gains justified. The different values were calculated for each subcorpus separately.

Length	Operation	Montreal	Toronto	Vancouver
25	$\geq$	11.126	12.180	12.730
50	$\geq$	13.909	14.808	15.240
75	$\geq$	17.114	18.000	18.321
mean	$\geq$	14.784	15.783	16.191
min	$\leq$	14.194	14.123	14.189
max	$\geq$	32.567	32.684	32.807

Table 5: Exact cutoffs for all length filters. Values were rounded to the nearest Integer.

Filter Category 3 - Amount: The fact that most content in user-generated corpora such as social media consists of highly active individuals has been noted (Doval et al., 2020; Tan et al., 2015). Subsampling approaches have been suggested in many different forms (Moreno-Ortiz and García-Gámez, 2023) with the standard process of combating the impact of skewed distributions being the subsampling of tweets from users to achieve a maximum amount of tweets per user (Miletic et al., 2021; Moreno-Ortiz and García-Gámez, 2023). Thus, I propose the following filters to investigate by limiting the number of tweets per user to 100 and 1000 which limits the impact of highly active users but also leaves a reasonable amount of data:

- $\leq$100 SS Tweets: Resulting corpus only contains users that have tweeted less than or exactly 100 times. Users that have tweeted more than that have their tweets randomly subsampled so that every user in the remaining corpus has exactly 100 tweets or less.
- $\leq$1000 SS Tweets: Resulting corpus only contains users that have tweeted less than or exactly 1000 times. Users that have tweeted more than that have their tweets randomly subsampled so that every user in the remaining corpus has exactly 1000 tweets or less.

Furthermore, an investigation into a specific activity span of users (low/mid/high frequency) was defined by setting up reasonable cut-offs for the specific frequency ranges where the expectation was that the range of activity positively correlates with the performance of the models:

- $<$100 Tweets: Resulting corpus only contains tweets from users that have tweeted less than 100 times in each subcorpus.
- $100 - 1000$ Tweets: Resulting corpus only contains tweets from users that have tweeted between 100 and 1000 times in each subcorpus.
- $>$1000 Tweets: Resulting corpus only contains tweets from users that have tweeted more than 1000 times in each subcorpus.

4.3.3 Combinations of Filters

All possible combinations of the filter categories were considered first. Because three filter categories were defined with multiple filter variants, there are many possible combinations. In order not to let the number of resulting models increase exponentially, only certain types were used for the combination of several filters. Therefore, the filters for the length category were limited to the minimum and maximum, and for the time category, partly due to the very low token counts of the first five years, I only decided on the tweets of the last five years for possible combinations. The reason for excluding all but two filter types for the length category was the fact that the token cutoffs and thus the results for this basic filter were very similar. It seemed logical to me to use the two most distinct filters (*Min, Max*) as the basis for combinations with other filter categories. It is worth noting that the combination of filters from the same category, e.g., below 100 tweets and above 1000 tweets, was not studied as this isn't an intersection but a union of sets, and including even more types of combinations would result in an abundance of models.

Table 6 shows all possible resulting models after combining filters. This results in 17 different combinations for the intersection of two filter categories and 10 different combinations for all intersection of filter categories. Together this results in 27 different combinations of filters. Altogether, the complete number of models to be investigated after including the unfiltered models, the models created by using one basic filter, the models created by combining two basic filters (combinations), and the models created by combining all basic filters, is 42 for each subcorpus.

Filter Category	Time	Length		Amount				
Filter Type	Last 5	Min	Max	<100	100-1000	>1000	>100 SS	>1000 SS
Last 5	-	×	×	×	×	×	×	×
Min		-		×	×	×	×	×
Max			-	×	×	×	×	×

Table 6: All possible combinations of filters investigated in this thesis. Valid combinations are marked with the × symbol.

4.4 Training

This section introduces the training architecture, its alignment, and the parameters of the training architecture.

4.4.1 Word Embeddings

As shown in (Tahmasebi et al., 2021; Schlechtweg et al., 2020; Kaiser et al., 2021) the field of semantic change detection is at the moment dominated by vector space models. The selected training architecture for this thesis is Skip-gram with negative sampling (**SGNS**) using the orthogonal procrutes alignment. SGNS achieves generally high scores across multiple tasks. (Tahmasebi et al., 2021; Schlechtweg et al., 2020). By choosing the orthogonal alignment of SGNS, models have to be trained for each subcorpus (Montreal, Toronto, Vancouver) and for each filter separately. To align the now separate models, they are length normalized, mean-centered, and finally aligned by computing an orthogonal constrained matrix Kaiser et al. (2021). Despite SGNS inherent instability Antoniak and Mimno (2018) only one model was trained for each filter instead of the standard approach of training multiple models and averaging over their evaluation results due to time constraints. I have used Rehurek and Sojka (2011) implementation of word2vec in gensim for this thesis due to its easy operability and cheap computational costs, allowing me to run multiple experiments without the need of rewriting and altering code. The training was done using 30 epochs for each model. All models were trained using the parameter configuration specified in Table 7.

4.4.2 Hyperparameters

Table 7 shows the hyperparameters used for SGNS. There weren't any other features used. The values of parameters are common standards or default values found in research (Tahmasebi et al., 2021; Gennaro et al., 2021) and the best-performing values for previous evaluation based on the CanEn dataset (Miletic et al., 2021). The comparatively low values for minimum occurrences of each word has to be set

to a low size as the counts of the test words in the evaluation set for semantic change detection, especially for the smaller models, are very low.

Parameter	Value
Dimensionality	100
Window Size	5
Minimum Occurrences	5
Negative Sampling	5
Learning Rate	0.025
Hierarchical Sampling	0

Table 7: Parameter configuration used for training of all models.

4.4.3 Configuration

This section shows the hardware specifications and the used external programming libraries for this thesis.

4.4.3.1 Programming Language and External Libraries

All coding-related tasks were accomplished using Python 3.10.8 and standard Unix commands when applicable. The handling and storage of the data was performed using *Pandas 1.3.5* Wes McKinney (2010). The preprocessing of the corpora was done using various modules of the *nltk* library Bird et al. (2009), including its Lemmatizer, WordNet, and TreebankWordTokenizer. The training of the models was done using *gensim 4.3.1.* Rehurek and Sojka (2011)

4.4.3.2 Hardware Specifications

The *Institut für Maschinelle Sprachverarbeitung* offers five different computing servers for students with different hardware specifications. Due to the sheer amount of models trained for this task, all servers were used during the preprocessing, training, and

evaluation steps of this thesis. Their CPU and RAM specifications can be found in Table 8.

Server Name	*CPU*	*Memory*
Nandu	AMD EPYC 7313 16-Core	258 GB
Kiwi	Intel(R) Xeon(R) CPU E5-2690 v4 @ 2.60GHz	512 GB
Dodo	AMD EPYC 7542 32-Core	514 GB
Phoenix	Intel(R) Xeon(R) CPU E5-2690 v2 @ 3.00GHz	256 GB
Strauss	AMD EPYC 7542 32-Core	1026GB

Table 8: Hardware specifications of the servers at the IMS.

4.5 Models

The following tables show the training time, the vocabulary size, tokens, and the total amount of tweets for the resulting models for all combinations after retaining minimum occurrences but before downsampling. Table 9 shows the parameters for the baseline and complete corpora while table 10 shows the parameters of the models for all basic filters. Tables 11, 12 and 13 show the parameters for all combinations of filters for all three corpora. 42 models were trained for each subcorpus, meaning there are 126 models in total. There is a significant difference in the input tokens between the three regional subcorpora with the Toronto and Vancouver corpora being much larger than the Montreal corpus for most filters.

4.5.1 Baseline and Full Corpus

Model - Filter	Corpus	Training Time	Vocabulary Size	Tokens	Tweets
Raw Corpus (No Preprocessing and no Filters)					
Raw	Montreal	62.67 min	1 300 044	332 386 439	23 469 526
Raw	Toronto	114.68 min	1 526 262	413 765 465	28 442 928
Raw	Vancouver	84.63 min	1 471 266	405 360 140	26 924 158
Full Corpus (Preprocessing and no Filters)					
Full	Montreal	227.88 min	318 267	203 426 612	23 463 669
Full	Toronto	91.67 min	336 840	253 078 070	28 436 274
Full	Vancouver	151.84 min	335 512	249 096 588	26 919 505

Table 9: Training time (in minutes), the vocabulary size, the number of tokens and sentences for the raw and full version of the three corpora.

4.5.2 Basic Filters

Model - Filter	Corpus	Training Time	Vocabulary Size	Tokens	Tweets
Amount of Tweets per User					
<100 Tweets	Montreal	2.63 min	47 541	8 699 192	1 074 921
100 – 1000 Tweets	Montreal	24.7 min	186 248	83 507 111	10 080 557
>1000 Tweets	Montreal	30.93 min	200 161	110 442 920	12 308 191
>100 SS Tweets	Montreal	42.1 min	113 904	37 771 859	4 548 261
>1000 SS Tweets	Montreal	179.97 min	279 290	163 023 913	19 147 645
Length					
25	Montreal	170.64 min	271 980	166 497 491	15 152 191
50	Montreal	136.13 min	242 790	141 858 868	11 706 310
75	Montreal	101.48 min	208 882	115 420 226	8 670 407
Mean Length	Montreal	124.97 min	232 022	133 177 153	10 647 398
Max Length	Montreal	20.51 min	80 984	26 115 701	1 169 123
Min Length	Montreal	79.1 min	151 027	61 014 244	11 757 359
Time					
First 5 Years	Montreal	6.62 min	35 517	5 352 641	666 790
Last 5 Years	Montreal	148.18 min	244 684	141 416 979	15 511 601
Amount of Tweets per User					
<100 Tweets	Toronto	3.3 min	38 427	7 256 282	823 180
100 – 1000 Tweets	Toronto	33.22 min	174 110	87 102 069	10 128 343
>1000 Tweets	Toronto	56.0 min	239 893	157 963 116	17 484 751
>100 SS Tweets	Toronto	28.45 min	104 083	38 602 407	4 421 224
>1000 SS Tweets	Toronto	105.77 min	285 108	193 560 977	22 053 155
Length					
25	Toronto	132.18 min	283 655	202 196 056	17 786 097
50	Toronto	109.23 min	251 750	170 807 700	13 704 417
75	Toronto	83.43 min	213 058	136 047 061	9 966 633
Mean Length	Toronto	103.61 min	239 549	159 549 218	12 420 877
Max Length	Toronto	20.87 min	86 565	31 347 381	1 403 540
Min Length	Toronto	57.33 min	152 731	70 803 792	13 408 423
Time					
First 5 Years	Toronto	4.17 min	39 633	6 958 472	836 825
Last 5 Years	Toronto	119.35 min	251 552	165 865 957	17 595 861
Amount of Tweets per User					
<100 Tweets	Vancouver	7.51 min	37 909	6 845 585	763 385
100 – 1000 Tweets	Vancouver	20.24 min	173 783	85 578 883	9 582 448
>1000 Tweets	Vancouver	171.5 min	238 675	155 923 166	16 573 672
>100 SS Tweets	Vancouver	15.72 min	103 907	37 979 146	4 207 456
>1000 SS Tweets	Vancouver	78.86 min	283 335	188 727 056	20 657 297
Length					
25	Vancouver	76.72 min	278 163	194 537 533	16 392 909
50	Vancouver	68.14 min	256 679	173 885 377	13 821 376
75	Vancouver	53.85 min	217 505	139 146 233	10 118 762
Mean Length	Vancouver	63.53 min	244 578	162 696 257	12 567 349
Max Length	Vancouver	10.9 min	89 239	33 076 778	1 467 194
Min Length	Vancouver	36.19 min	145 102	64 060 173	11 822 058
Time					
First 5 Years	Vancouver	4.03 min	44 126	8 678 021	1 032 460
Last 5 Years	Vancouver	65.11 min	249 204	160 263 633	16 211 131

Table 10: Training time (in minutes), the amount of word types, tokens and total tweets for all basic filters across the three corpora.

4.5.3 Combinations of Filters

Model - Filter	Corpus	Training Time	Vocabulary Size	Tokens	Tweets
Combination - Length and Amount					
Max Length × <100 Tweets	Montreal	0.4 min	16 166	1 503 159	72 989
Max Length ×100 − 1000 Tweets	Montreal	2.12 min	46 005	9 392 569	433 337
Max Length × >1000 Tweets	Montreal	3.42 min	56 900	15 030 148	662 797
Max Length × >100 SS Tweets	Montreal	3.82 min	62 664	16 520 955	763 689
Max Length × >1000 SS Tweets	Montreal	5.89 min	80 497	25 804 961	1 159 442
Min Length × <100 Tweets	Montreal	1.51 min	24 058	2 876 934	608 867
Min Length ×100 − 1000 Tweets	Montreal	11.5 min	92 639	27 235 151	5 346 444
Min Length × >1000 Tweets	Montreal	12.2 min	94 471	30 521 013	5 802 048
Min Length × >100 SS Tweets	Montreal	8.06 min	75 874	19 486 231	3 792 193
Min Length × >1000 SS Tweets	Montreal	24.88 min	149 661	60 173 289	11 598 082
Combination - Length and Time					
Max Length × Last 5 Years	Montreal	4.76 min	80 911	26 083 912	1 166 459
Min Length × Last 5 Years	Montreal	15.29 min	110 616	38 679 659	7 473 483
Combination - Time and Amount					
Last 5 Years × <100	Montreal	3.98 min	41 750	7 822 269	868 486
Last 5 Years × 100-1000	Montreal	16.51 min	143 467	57 461 564	6 663 372
Last 5 Years × >1000	Montreal	17.58 min	155 131	76 141 933	7 979 945
Last 5 Years × >100 SS	Montreal	8.55 min	104 450	34 105 403	3 984 009
Last 5 Years × >1000 SS	Montreal	33.58 min	221 702	119 237 544	13 373 479
Combination - Length and Time and Amount					
Max Length × Last 5 Years × <100 Tweets	Montreal	0.38 min	16 164	1 502 357	72 911
Max Length × Last 5 Years × 100-1000 Tweets	Montreal	2.04 min	45 955	93 77 990	432 014
Max Length × Last 5 Years × >1000 Tweets	Montreal	3.04 min	56 850	15 013 923	661 534
Max Length × Last 5 Years × >100 SS Tweets	Montreal	3.5 min	62 488	16 490 844	761 134
Max Length × Last 5 Years × >1000 SS Tweets	Montreal	5.31 min	80 454	25 773 372	1 156 778
Min Length × Last 5 Years × <100 Tweets	Montreal	1.18 min	20 322	2 256 156	483 055
Min Length × Last 5 Years × 100-1000 Tweets	Montreal	7.02 min	67 863	17 343 513	3 432 059
Min Length × Last 5 Years × >1000 Tweets	Montreal	7.82 min	69 721	18 795 848	3 558 369
Min Length × Last 5 Years × >100 SS Tweets	Montreal	6.66 min	65 115	15 858 055	3 101 349
Min Length × Last 5 Years × >1000 SS Tweets	Montreal	15.53 min	110 090	38 436 944	7 431 156

Table 11: Training time (in minutes), the amount of word types, tokens and total tweets for all combinations of filters of the Montreal corpus.

Model - Filter	Corpus	Training Time	Vocabulary Size	Tokens	Tweets
Combination - Length and Amount					
Max Length × <100 Tweets	Toronto	0.4 min	15 756	1 529 378	73 773
Max Length ×100 − 1000 Tweets	Toronto	1.97 min	46965	10 582 833	486 828
Max Length × >1000 Tweets	Toronto	3.49 min	63 840	19 042 061	842 939
Max Length × >100 SS Tweets	Toronto	3.6 min	65 637	19 398 876	895 513
Max Length × >1000 SS Tweets	Toronto	5.65 min	85 604	30 715 776	1 382 142
Min Length × <100 Tweets	Toronto	1.18 min	17 497	1 989 577	413 238
Min Length ×100 − 1000 Tweets	Toronto	10.16 min	81 980	25 976 280	5 032 692
Min Length × >1000 Tweets	Toronto	15.94 min	110 583	42 486 254	7 962 493
Min Length × >100 SS Tweets	Toronto	7.57 min	69 171	19 501 378	3 705 691
Min Length × >1000 SS Tweets	Toronto	26.27 min	150 691	69 553 381	13 174 392
Combination - Length and Time					
Max Length × Last 5 Years	Toronto	6.39 min	86 476	31 308 735	1 400 340
Min Length × Last 5 Years	Toronto	16.32 min	108 146	41 670 630	7 903 455
Combination - Time and Amount					
Last 5 Years× <100	Toronto	1.58 min	34 809	6 251 851	690 198
Last 5 Years× 100-1000	Toronto	14.37 min	133 772	58 886 817	6 541 474
Last 5 Years× >1000	Toronto	24.71 min	177 148	100 150 916	10 364 189
Last 5 Years× >100 SS	Toronto	9.36 min	97 091	35 540 391	3 948 044
Last 5 Years× >1000 SS	Toronto	35.75 min	224 180	137 765 968	14 899 535
Combination - Length and Time and Amount					
Max Length × Last 5 Years × <100 Tweets	Toronto	0.4 min	15 752	1 528 702	73 703
Max Length × Last 5 Years × 100-1000 Tweets	Toronto	1.96 min	46 917	10 569 635	485 667
Max Length × Last 5 Years × >1000 Tweets	Toronto	3.5 min	63 786	19 017 523	840 970
Max Length × Last 5 Years × >100 SS Tweets	Toronto	3.57 min	65 684	19 368 704	892 413
Max Length × Last 5 Years × >1000 SS Tweets	Toronto	5.7 min	85 538	30 678 098	1 378 946
Min Length × Last 5 Years × <100 Tweets	Toronto	1.03 min	15 106	1 613 328	338 463
Min Length × Last 5 Years × 100-1000 Tweets	Toronto	6.5 min	60 382	16 105 634	3 143 051
Min Length × Last 5 Years × >1000 Tweets	Toronto	8.87 min	77 379	23 696 290	4 421 941
Min Length × Last 5 Years × >100 SS Tweets	Toronto	6.37 min	60 195	15 985 287	3 052 556
Min Length × Last 5 Years × >1000 SS Tweets	Toronto	15.5 min	107 397	41 375 963	7 854 248

Table 12: Training time (in minutes), the amount of word types, tokens and total tweets for all combinations of filters of the Toronto corpus.

Model - Filter	Corpus	Training Time	Vocabulary Size	Tokens	Tweets
Combination - Length and Amount					
Max Length × <100 Tweets	Vancouver	0.8 min	15 485	1 414 750	68 270
Max Length ×100 − 1000 Tweets	Vancouver	3.63 min	47 094	10 461 126	478 988
Max Length × >1000 Tweets	Vancouver	7.42 min	66 779	21 005 402	919 936
Max Length × >100 SS Tweets	Vancouver	6.27 min	66 506	19 428 213	891 943
Max Length × >1000 SS Tweets	Vancouver	10.38 min	88 483	32 391 204	1 443 117
Min Length × <100 Tweets	Vancouver	1.97 min	16 934	1 816 423	371 194
Min Length ×100 − 1000 Tweets	Vancouver	15.81 min	78 832	23 509 312	4 422 899
Min Length × >1000 Tweets	Vancouver	24.86 min	104 307	38 399 732	7 027 965
Min Length × >100 SS Tweets	Vancouver	12.66 min	68 142	18 739 688	3 478 211
Min Length × >1000 SS Tweets	Vancouver	40.86 min	143 469	63 000 421	11 626 817
Combination - Length and Time					
Max Length × Last 5 Years	Vancouver	10.99 min	89 160	33 042 869	1 464 397
Min Length × Last 5 Years	Vancouver	23.52 min	101 365	36 320 278	6 697 604
Combination - Time and Amount					
Last 5 Years× <100	Vancouver	2.91 min	33 719	5 669 598	610 558
Last 5 Years× 100-1000	Vancouver	22.24 min	130 773	55 580 526	5 894 151
Last 5 Years× >1000	Vancouver	33.75 min	177 022	98 448 909	9 706 422
Last 5 Years× >100 SS	Vancouver	13.01 min	96 618	34 478 717	3 681 505
Last 5 Years× >1000 SS	Vancouver	45.27 min	221 571	131 746 034	13 588 779
Combination - Length and Time and Amount					
Max Length × Last 5 Years × <100 Tweets	Vancouver	0.6 min	15 481	1 414 062	68 210
Max Length × Last 5 Years × 100-1000 Tweets	Vancouver	3.24 min	47 058	10 447 135	477 790
Max Length × Last 5 Years × >1000 Tweets	Vancouver	6.33 min	66 726	20 986 421	918 397
Max Length × Last 5 Years × >100 SS Tweets	Vancouver	5.93 min	66 538	19 397 518	889 312
Max Length × Last 5 Years × >1000 SS Tweets	Vancouver	9.87 min	88 397	32 357 676	1 440 320
Min Length × Last 5 Years × <100 Tweets	Vancouver	1.56 min	14 255	1 389 032	287 292
Min Length × Last 5 Years × 100-1000 Tweets	Vancouver	8.85 min	56 218	13 678 477	2 586 933
Min Length × Last 5 Years × >1000 Tweets	Vancouver	13.64 min	72 971	21 014 075	3 823 379
Min Length × Last 5 Years × >100 SS Tweets	Vancouver	9.87 min	58 276	14 818 509	2 760 216
Min Length × Last 5 Years × >1000 SS Tweets	Vancouver	23.32 min	100 742	36 046 639	6 653 535

Table 13: Training time (in minutes), the amount of word types, tokens and total tweets for all combinations of filters of the Vancouver corpus.

4.6 Evaluation

This section focuses on the last step of the experimental setup, namely the evaluation of the learned models based on word similarity and semantic change detection.

4.6.1 Word Similarity

As mentioned in section 2, the evaluation of word similarity is done by calculating the correlation score between the predicted measurement and the actual, human-annotated measurements. Deriving the predicted measurement consists of calculating the cosine distance for the vectors of two words whose similarity is to be assessed in the datasets. There are four test sets used in this thesis as described in section section 4.2.3. To combat any eventual outliers and to counteract the fact that some words may not appear in the corpora, the average correlation score for both, Pearson and Spearman correlation, of all four selected test sets will be used to compare the results across different models.

4.6.2 Semantic Change Detection

The evaluation of the learned word vectors is done using the CanEn test set created specifically for this corpus. The task is defined as a binary classification task in which half of the words expected to change their semantic meaning, which is reflected in regional patterns, hence the comparison of regional subcorpora, are put into one class and the other half is defined as a control instance where no change in semantic meaning is expected. This is similar to the description in Schlechtweg et al. (2020) for evaluating word embeddings for semantic change detection.

The test set contains 80 words with 40 words in each class. The standard process here is now the ranking of the words according to a distance measure (here: cosine distance), and the subsequent assignment into one of the two classes depending on the score. The distances are computed by comparing the resulting word

vectors of each direct comparison between the same word across the three subcorpora: MT, MV, TV. Thus, for the MT score, the cosine distance between the word embedding for the respective word in the Montreal and the Vancouver corpus is measured. The top 40 words, whose regionally specific vectors are the most distant across different corpora, are then assigned to the class where a change of the semantic meaning was to be expected. The accuracy score is accordingly computed by measuring how many words are in the correct class. It is important that the word to be examined actually occurs in all three subcorpora. If this is not the case, the word is removed from the test set, and the original 40-40 split is adjusted so that half of the words are always in the respective groups. This is especially important for models that already have a low-frequency count of the investigated words. The sum and the average for each of the comparisons of corpora are also accordingly calculated and reported.

Furthermore, the accuracy scores are additionally calculated for three further metrics as proposed by Miletic et al. (2021). These are the average difference between the MT and MV distances (5), the difference between this and the TV distance (6), and the ratio between the two values (7).

$$avg(MT, MV) = \frac{MT + MV}{2} \tag{5}$$

$$diff((MT, MV), TV) = avg(MT, MV) - TV \tag{6}$$

$$ratio((MT, MV), TV) = \frac{avg(MT, MT)}{TV} \tag{7}$$

This results in six different metrics that attempt to determine the extent to which the expected semantic changes have actually occurred.

5 Results and Discussion

This section will present the results of all filters and test sets for (i) word similarity and (ii) semantic change detection. A distinction is made between the respective filters themselves and their combinations. Any outliers, findings, and other results are addressed, and the discussion is supported with visual graphs and tables.

5.1 Word Similarity

To reiterate the evaluation for word similarity, the mean correlation coefficient over all four previously mentioned test sets is calculated and will be presented here.

5.1.1 Results per Filter

Filter Category	Filter Type	Combinations	Montreal	Toronto	Vancouver	Average
Time	First 5 Years	1	0.520/0.526	0.545/0.552	0.543/0.552	0.536/0.545
	Last 5 Years	1	*0.610/0.611*	*0.603/0.604*	*0.622/0.624*	*0.612/0.613*
Tweet Length	25	1	0.612/0.613	***0.624/0.626***	0.627/0.627	0.621/0.622
	50	1	0.608/0.610	0.614/0.618	0.625/0.624	0.616/0.617
	75	1	***0.612/0.623***	0.618/0.620	*0.627/0.630*	*0.621/0.624*
	min	1	0.586/0.592	0.590/0.593	0.596/0.598	0.591/0.594
	max	1	0.603/0.607	0.591/0.591	0.610/0.612	0.601/0.603
	mean	1	0.612/0.615	0.612/0.615	0.627/0.628	0.617/0.619
Amount of Tweets	<100	1	0.554/0.561	0.546/0.552	0.565/0.573	0.555/0.562
	100-1000	1	0.605/0.607	0.613/0.618	0.617/0.619	0.612/0.614
	>1000	1	0.615/0.615	*0.615/0.619*	*0.622/0.622*	*0.620/0.619*
	>100 SS	1	0.594/0.601	0.604/0.607	0.617/0.618	0.605/0.608
	>1000 SS	1	*0.616/0.616*	0.613/0.615	0.618/0.620	0.617/0.617
No Filtering	Full Corpus	1	*0.617/0.621*	*0.624/0.625*	***0.628/0.629***	***0.623/0.625***
No Filtering	Raw Corpus	1	0.539/0.536	0.543/0.536	0.552/0.546	0.545/0.539

Table 14: Average Pearson correlation coefficient r / Spearman correlation coefficient ρ for each filter and corpus for *Word Similarity* across all test sets. Italic values represent the highest value of the column across the same filter category. Bold and italic values represent the highest value of the column.

Table 14 shows the complete results for all basic filters for word similarity averaged over all test sets while table 15 shows the same but also additionally the average results for all filters that participate in combinations of filters. It should be clear from first glance that the filters used in combination with other filters produce on average significantly worse results than filters that have not been used in combinations. The raw baseline corpus performs worst, which validates the importance of the preprocessing steps, especially in contrast to the full baseline corpus.

If we first look at table 14 and its presented results, we see that for most filters, the correlation scores are lower than for the full corpus. Although the complete corpus gives the best results on average with an average correlation score of 0.625, some individual filters give quite competitive, if not only minimally worse results. For instance, the filters that trim the tweets to a certain minimum length to be reached achieve results very similar to those of the original corpus. For example, for the filters that restrict the length to minimally 25% (lower quartile) or 75% (upper quartile) of the tweet length, the correlations reach an average value of 0.622. Also, the filters that filter tweets by the last 5 years and by over 1000 tweets per user also achieve very good results.

Let's start with the individual filter categories. For the time category, it becomes apparent that the first five years of tweets deliver significantly worse results than the last five years. This can most likely be explained by the massively lower number of tweets and the resulting significantly lower number of tokens (a couple of millions compared to hundreds of millions) in the earlier years compared to the later years. For the category of length, it can be seen that all filters apart from the minimum and maximum yield similarly good results, while for the category of the amount of tweets, the larger number of tweets per user leads to better results on average. This is particularly evident since the filter that considers users with 100 tweets or less achieves the worst results in this category.

There is, somewhat surprisingly, a stark difference in the best results across the regional subcorpora. The model trained on the Toronto and Vancouver corpora achieve, respectively, better results than the models trained on the Montreal corpus despite using the same filtering method. While there is a significant difference in token count for multiple filters across the corpora, there seems to be an inherent difference in the data quality for the regional subcorpora.

5.1.2 Results per Filter Including Combinations

Filter Category	Filter Type	Combinations	Montreal	Toronto	Vancouver	Average
Time	Last 5 Years	18	*0.567/0.570*	*0.567/0.569*	*0.579/0.581*	*0.574/0.573*
Tweet Length	min	12	0.552/0.559	0.551/0.556	0.560/0.564	0.554/0.557
	max	12	*0.570/0.570*	*0.570/0.570*	*0.580/0.581*	*0.572/0.572*
Amount of Tweets	<100	6	0.494/0.498	0.482/0.483	0.492/0.494	0.488/0.484
	100-1000	6	0.578/0.583	0.581/0.587	0.589/0.592	0.583/0.588
	>1000	6	0.583/0.588	0.587/0.590	0.599/0.604	0.590/0.594
	>100 SS	6	0.574/0.580	0.580/0.581	0.593/0.597	0.582/0.586
	>1000 SS	6	*0.595/0.599*	*0.598/0.600*	*0.608/0.610*	*0.601/0.603*
No Filtering	Full Corpus	1	***0.617/0.621***	***0.624/0.625***	***0.628/0.629***	***0.623/0.625***
No Filtering	Raw Corpus	1	0.539/0.536	0.543/0.536	0.552/0.546	0.545/0.539

Table 15: Average Pearson correlation coefficient r / Spearman Correlation coefficient ρ for each filter and corpus for filters that appear in more than one combination across all test sets. Italic values represent the highest value of the column across the same filter category. Bold and italic values represent the highest value of the column.

Examining table 15 and its presented results, we notice that the results of the filters involved in multiple combinations have significantly decreased compared to just using the individual filters. And although this does not generally affect the order of the results of the individual filtering types per filtering category in particular, certain differences from the previously discussed table can be seen. For example, the ranking of the results of the filtering category amount is noticeably changed. Especially the filters that use subsampling techniques achieve surprisingly good values on average

so subsampling users with more than 1000 tweets on average is the best-performing filter in this category with an average correlation score of 0.602.

Furthermore, it should be said that the filter type which only includes users that tweet less than 100 times has by far the worst results. While this performance was similarly reported in the first table, it is noticeably worse than all other filters, and even worse than just using the raw, unpreprocessed corpus, which largely stems from the fact that the models using this filter have by far the lowest token count as shown in section 4.5. There are only small differences between the score for the two filter types for length, which continues the trend from the first table, that max performs better than min.

The previously noticed trend that the different regional subcorpora achieve different results is continued here in a slight alteration. The models trained on the Montreal and the Toronto subcorpora achieve closer results to each other while the models trained on the Vancouver corpora have, on average, better results. An important key message from these findings lies in the results of the < 100 filter, which achieves the best results for the Montreal corpus in combination with other filters. Judging by the token counts, the models trained on this filter and corpus have more tokens than the models trained on the same filter for the other corpora. There seems to be a significant difference in how Twitter is used across the different regions with Montreal having more people use Twitter that write less than a hundred tweets.

In the next section, the full distribution of the results will be discussed.

5.1.3 Analysis

Table 16 shows the difference between the Pearson correlation coefficient scores for the individual filters and the combinations by subtracting the two values from each other and dividing the result by the number of times the respective filter was used in a combination with other filters with the last row representing the average drop in correlation score per combination for each filter.

Filter	Last 5	Min	Max	<100	100-1000	>1000	>100 SS	>1000 SS
Individual Correlation	0.612	0.591	0.601	0.555	0.612	0.620	0.605	0.617
Average of Correlation of Combinations	0.574	0.554	0.572	0.488	0.583	0.590	0.582	0.601
Amount of Combinations	18	12	12	6	6	6	6	6
Drop	0.038	0.037	0.029	0.067	0.029	0.03	0.023	0.016
Drop per Combination	0.002	0.003	0.002	0.01	0.005	0.005	0.004	0.002

Table 16: Overview of the difference of the average Pearson coefficient r when filters are used once compared to their average usage in all combinations.

There is noticeable a drop in performance across the board. This drop is especially noticeable for the <100 filter where the score drops by 0.067. On average, the filters lose a non-negligible percentage of their score. However, there is an interesting finding and that is that certain filters do not necessarily get much worse per combination despite the large number of combinations. Thus, the filters of the last 5 years, length, and >1000SS lose just under 0.002 per additional combination. This leads to the fact that these filters are quite stable and indicates that they can likely deliver solid results in even more complex filtering processes involving even further filtering categories.

Figure 8 shows a boxplot for all correlation scores of filters used in combinations with other filters. All filters, with the exception of the <100 filter are quite stable and only have outliers in combination with the <100 filter. However, since the filters of the same category were not combined with each other, this statement cannot be fully generalized. Nevertheless, the result is unlikely to change since adding the <100 filter to the other filters of this category only adds context as it is impossible to intersect different variants of the same category and the union of variants of the same filter category only adds data, and therefore should not change the result significantly.

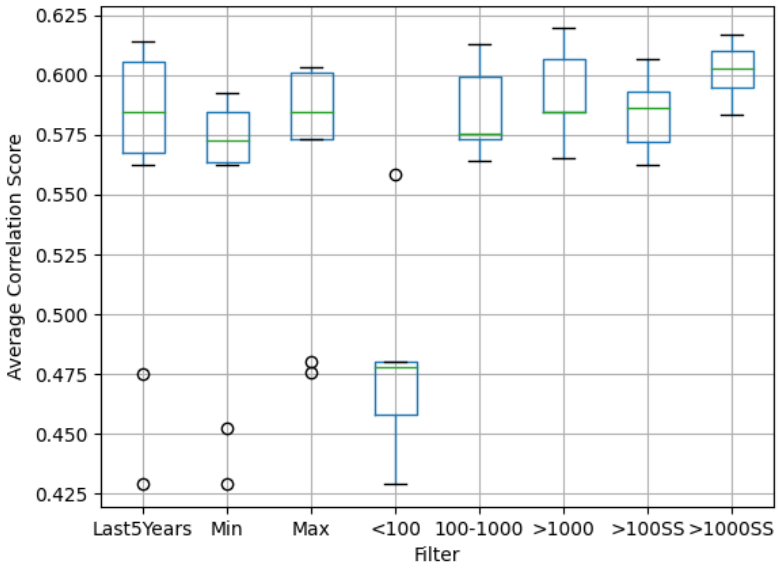

Figure 8: Boxplot of all average correlation scores for all filters that have been used more than once in combinations.

Figure 9 shows a scatterplot of the number of tokens from the tables in section 4.5 and the correlation scores from the previously explained tables. Ultimately, one should always bear in mind that the number of tokens has an immense influence on the results of either filtering method, the individual filters, and their combinations (Wendlandt et al., 2018).

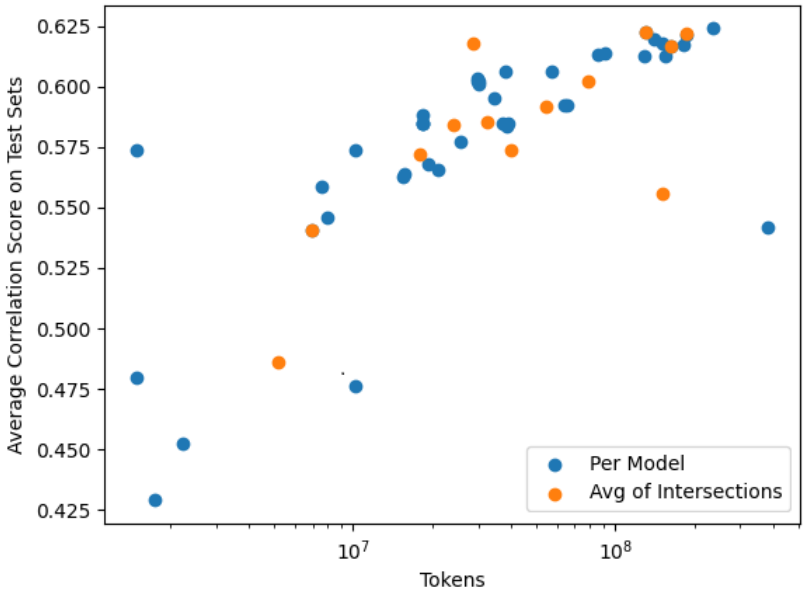

Figure 9: Scatterplot of all models and their combinations showing the average correlation score on the task of word similarity using a logarithmic scale on the x-axis depicting the magnitude of the token count.

The figure shows the correlation between the token count and the resulting correlation score for each (i) model in blue and the average scores for the (ii) basic filters in orange. The X-axis is displayed logarithmically to better show the differences in the number of tokens between the different models. At one end, we have models with just barely above one million tokens, and at the other end the unpreprocessed corpus with several hundred million tokens, in the case of the raw corpus around 400 million. There is a clear correlation between the number of tokens and the performance

of the respective models, which has previously been reported for embeddings in word similarity (Antoniak and Mimno, 2018; Wang et al., 2019). Calculating Spearman's correlation returns the correlation coefficient $\rho = 0.808$ for all models and $\rho = 0.659$ for the basic filters, which indicates a strong correlation.

However, there is a very important finding here. It is visible in the graph that some models already achieve very good results with only a fraction of the original corpus size. For example, the models with only $\approx 10^7$ tokens already achieve similar results to the completely unprocessed corpus with more than $4 \cdot 10^8$ tokens. It is also important to note that although the complete corpus delivers the best result, other models with far fewer tokens already deliver comparable results. This is particularly fascinating because the results increase as the number of tokens increases, but they do not increase significantly.

It seems that there is a hard limit of about 0.625, which cannot be exceeded despite the large amount of training data. This may mean that this is the maximum possible result of this specific corpus and training architecture. Word embeddings from social media corpora lead to different results than from other, more formal sources (Tan et al., 2015; Elekes et al., 2017; Doval et al., 2020). Due to the structure of the evaluation process of this thesis, the comparison with results in research isn't inherently simple. That said, Doval et al. (2020) achieve similar results using noisy Twitter data and word2vec on the WordSim353 and SimLex999 evaluation sets with a Spearman correlation of ≈ 0.65.

Further tables, presenting the complete correlation scores for each filter, subcorpora, and evaluation set can be found in the appendix under section 7.1, which is divided into all possible filtering options and combinations of filters.

5.2 Semantic Change Detection

As presented in detail in section 4.6.2, the evaluation process for semantic change detection consists of calculating the accuracy scores for a binary class prediction task by sorting the values for a direct comparison between two subcorpora and splitting them in half. From this, we create further metrics and calculate their accuracy score the same way.

5.2.1 Results per Filter

Filter Category	Filter Type	Combinations	MT	MV	TV	Avg	Dist	Diff	Ratio
Time	First 5 Years	1	0.486	0.459	*0.514*	0.486	0.486	0.486	0.405
	Last 5 Years	1	*0.525*	*0.550*	0.475	*0.517*	*0.525*	0.500	*0.562*
Tweet Length	25	1	***0.575***	0.525	*0.525*	***0.542***	***0.575***	0.600	0.588
	50	1	0.525	0.500	0.475	0.500	0.500	0.600	0.588
	75	1	0.500	0.525	*0.525*	0.517	0.500	0.600	0.562
	min	1	0.550	0.500	0.475	0.508	0.525	0.575	0.538
	max	1	0.425	0.525	0.400	0.450	0.450	0.525	0.588
	mean	1	0.500	*0.550*	0.450	0.500	0.525	*0.625*	*0.625*
Amount of Tweets	<100	1	*0.513*	0.513	0.461	0.496	0.487	0.539	0.510
	100-1000	1	0.500	0.525	0.425	0.483	0.500	*0.625*	*0.550*
	>1000	1	0.500	***0.575***	***0.575***	*0.525*	*0.525*	0.575	*0.550*
	>100 SS	1	0.450	0.475	***0.575***	0.475	0.500	0.550	0.450
	>1000 SS	1	0.500	***0.575***	***0.575***	*0.525*	*0.525*	0.575	*0.550*
No Filtering	Full Corpus	1	*0.550*	***0.575***	*0.450*	*0.525*	*0.550*	0.525	***0.650***
	Raw Corpus	1	0.525	0.500	0.425	0.483	0.525	***0.675***	0.562

Table 17: Accuracy score for each filter and corpus using the CanEn test set and a standard 50/50 split and no combinations for *Semantic Change Detection*. Italic values represent the highest value of the column across the same filter category. Bold and italic values represent the highest value of the column.

Table 17 presents the accuracy scores for each metric for all basic filters and regional subcorpora comparisons. To reiterate the respective columns, the first three columns depict the accuracy scores for direct comparisons between two regional subcorpora (e.g. MT being the distance of the word vectors between the same word in the Montreal and the Toronto corpus). In the column with the title average, the accuracy scores of the previous three columns are averaged. The last three metrics evaluate the average distance between the MT + MV distance, the difference between the previous value and the TV distance and the ratio between these two values, respectively.

It can be seen in the table that the results for the accuracy of the respective metrics and direct comparisons of the individual regional subcorpora range between ≈ 0.4 and ≈ 0.6. This means that there is a wide difference between the worst-performing and the best-performing filters.

If we focus on each individual filter category first, we can see that the first 5 years filter has the highest accuracy scores for TV, but relatively lower accuracy for the other two comparisons. This is the complete opposite of the last 5 years filter, which performs much better for the comparison where Montreal is involved. The filters of the length category have, similar to their results for word similarity, once again good comparable results to the full corpus, which is again the best-performing model. As for the filters that focus on the amount of tweets per user, >1000 and >1000SS filters consistently show higher accuracy across most of the comparisons, indicating that having more tweets in the dataset contributes to better semantic change detection ad that controlling for highly productive users (by subsampling as opposed to using the full corpus) improves performance in some corpus settings. The best-performing filter outside of the complete corpus is the 25% filter for length and the aforementioned filters for the amount of tweets.

5.2.2 Results per Filter Including Combinations

Table 18 shows the accuracy scores for each metric for all filters that have been investigated in combinations with other filters. By far the most obvious and yet most surprising result is that the filter for < 100 achieves the best result. This is especially shocking as this is the filter with the lowest amount of tokens due to its inherent property of only including tweets from users with less than one hundred tweets. After that, the > 1000 subsample filter achieves the best scores. This trend has also been seen in the previous table, making this filter a good selection for this particular task. Overall the scores are dropping by combining multiple filters, which was also reported on word similarity.

Filter Category	Filter Type	Combinations	MT	MV	TV	Avg	Dist	Diff	Ratio
Time	Last 5 Years	18	*0.493*	*0.504*	*0.456*	*0.484*	*0.496*	*0.519*	*0.528*
Tweet Length	min	12	*0.524*	*0.499*	*0.473*	*0.499*	*0.519*	*0.534*	0.521
	max	12	0.460	0.489	0.440	0.463	0.473	0.497	*0.537*
Amount of Tweets	<100	6	***0.571***	0.507	***0.477***	*0.518*	*0.540*	0.480	0.533
	100-1000	6	0.483	0.492	0.442	0.472	0.475	0.546	0.511
	>1000	6	0.476	0.522	0.451	0.483	0.498	*0.548*	0.529
	>100 SS	6	0.460	0.490	***0.477***	0.476	0.483	0.483	0.512
	>1000 SS	6	0.492	*0.525*	0.450	0.489	0.496	0.546	*0.548*
No Filtering	Full Corpus	1	*0.550*	***0.575***	*0.450*	***0.525***	***0.550***	0.525	***0.650***
	Raw Corpus	1	0.525	0.500	0.425	0.483	0.525	***0.675***	0.562

Table 18: Accuracy score for each filter and corpus using the CanEn test set and a standard $n - n$ split for *Semantic Change Detection*. Italic values represent the highest value of the column across the same filter category. Bold and italic values represent the highest value of the column.

An important consideration is the fact that some of these filters have, on average, an accuracy score for some metrics below 0.5 on a balanced evaluation set. This means, that essentially the model's prediction relies on chance. The three additional metrics seem to be more stable compared to the individual comparisons as they ultimately have accuracy scores above the 0.5 threshold.

Table 19 and table 20 show the complete results, including the actual split, for all filters. Table 31 and table 32 in the appendix present the best possible scores by altering the bisection to an unequal split and calculating the highest possible sum of all three comparisons. One possible explanation for the good performance of the < 100 filter may be the low number of occurring words of the evaluation set in the three corpora. Thus, one can see in table 20 that many of the results of the combinations using the < 100 filter are based on a very low number of actually occurring words and therefore skew the result in a different direction. Combining, for example, the min, < 100, and the last 5 years filters results in only 38 test cases.

$N - N$ Split for all Basic Filters

Model - Filter	Split	MT	MV	TV	Sum	Avg	Dist	Diff	Ratio
Raw Corpus (No Preprocessing and no Filters)									
Raw	40	0.525	0.500	0.425	1.450	0.483	0.525	0.675	0.562
Full Corpus (Preprocessing but No Filters)									
Full	40	0.550	0.575	0.450	1.575	0.525	0.550	0.525	0.650
Amount of Tweets per User									
<100 Tweets	38	0.513	0.513	0.461	1.487	0.496	0.487	0.539	0.513
100 – 1000 Tweets	40	0.500	0.525	0.425	1.450	0.483	0.500	0.625	0.550
>1000 Tweets	40	0.500	0.575	0.500	1.575	0.525	0.525	0.575	0.550
>100 SS Tweets	40	0.450	0.475	0.500	1.425	0.475	0.500	0.550	0.450
>1000 SS Tweets	40	0.500	0.575	0.500	1.575	0.525	0.525	0.575	0.550
Length									
25	40	0.575	0.525	0.525	1.625	0.542	0.575	0.600	0.588
50	40	0.525	0.500	0.475	1.500	0.500	0.500	0.600	0.588
75	40	0.500	0.525	0.525	1.550	0.517	0.500	0.600	0.562
Mean Length	40	0.500	0.550	0.450	1.500	0.500	0.525	0.625	0.625
Max Length	40	0.425	0.525	0.400	1.350	0.450	0.450	0.525	0.588
Min Length	40	0.550	0.500	0.475	1.525	0.508	0.525	0.575	0.538
Time									
First 5 Years	37	0.486	0.459	0.514	1.459	0.486	0.486	0.486	0.405
Last 5 Years	40	0.525	0.550	0.475	1.550	0.517	0.525	0.500	0.562

Table 19: Accuracy score, their sum, and mean for the three individual comparisons for all basic filters. The columns after that indicate the accuracy score for the average distance of the sum of MT and MV compared to TV, the difference between MT and MV compared to TV, and the ratio between the two groups.

$N - N$ Split for all Combinations of Filters

Model - Filter	Split	MT	MV	TV	Sum	Avg	Dist	Diff	Ratio
Combination - Length and Amount									
Max Length × <100 Tweets	27	0.611	0.426	0.463	1.500	0.500	0.537	0.463	0.519
Max Length ×100 − 1000 Tweets	40	0.500	0.500	0.475	1.475	0.492	0.500	0.55	0.525
Max Length × >1000 Tweets	39	0.43	0.532	0.43	1.392	0.464	0.506	0.506	0.506
Max Length × >100 SS Tweets	40	0.475	0.475	0.450	1.400	0.467	0.475	0.450	0.562
Max Length × >1000 SS Tweets	40	0.400	0.500	0.450	1.350	0.450	0.400	0.500	0.575
Min Length × <100 Tweets	23	0.702	0.532	0.489	1.723	0.574	0.617	0.447	0.574
Min Length ×100 − 1000 Tweets	40	0.500	0.500	0.425	1.425	0.475	0.475	0.575	0.500
Min Length × >1000 Tweets	40	0.450	0.525	0.450	1.425	0.475	0.500	0.500	0.512
Min Length × >100 SS Tweets	40	0.475	0.500	0.575	1.550	0.517	0.500	0.475	0.475
Min Length × >1000 SS Tweets	40	0.475	0.475	0.450	1.400	0.467	0.500	0.600	0.562
Combination - Length and Time									
Max Length × Last 5 Years	40	0.450	0.475	0.425	1.350	0.450	0.475	0.525	0.588
Min Length × Last 5 Years	40	0.525	0.500	0.500	1.525	0.508	0.525	0.625	0.488
Combination - Time and Amount									
Last 5 Years × <100	37	0.533	0.533	0.533	1.600	0.533	0.533	0.427	0.533
Last 5 Years × 100-1000	40	0.475	0.475	0.475	1.425	0.475	0.450	0.575	0.538
Last 5 Years × >1000	40	0.500	0.575	0.475	1.550	0.517	0.500	0.600	0.538
Last 5 Years × >100 SS	40	0.500	0.500	0.450	1.450	0.483	0.475	0.450	0.512
Last 5 Years × >1000 SS	40	0.525	0.575	0.425	1.525	0.508	0.525	0.575	0.575
Combination - Length and Time and Amount									
Max Length × Last 5 Years × <100 Tweets	27	0.426	0.500	0.426	1.352	0.451	0.426	0.463	0.519
Max Length × Last 5 Years × 100-1000 Tweets	40	0.450	0.475	0.425	1.350	0.450	0.425	0.450	0.450
Max Length × Last 5 Years × >1000 Tweets	39	0.456	0.481	0.43	1.367	0.456	0.481	0.557	0.519
Max Length × Last 5 Years × >100 SS Tweets	40	0.425	0.475	0.425	1.325	0.442	0.500	0.475	0.562
Max Length × Last 5 Years × >1000 SS Tweets	40	0.475	0.500	0.425	1.400	0.467	0.500	0.500	0.525
Min Length × Last 5 Years × <100 Tweets	19	0.641	0.538	0.487	1.667	0.556	0.641	0.538	0.538
Min Length × Last 5 Years × 100-1000 Tweets	40	0.475	0.475	0.425	1.375	0.458	0.500	0.500	0.500
Min Length × Last 5 Years × >1000 Tweets	40	0.525	0.450	0.425	1.400	0.467	0.475	0.55	0.55
Min Length × Last 5 Years × >100 SS Tweets	40	0.450	0.450	0.475	1.375	0.458	0.450	0.500	0.512
Min Length × Last 5 Years × >1000 SS Tweets	40	0.525	0.55	0.500	1.575	0.525	0.525	0.525	0.500

Table 20: Accuracy score, their sum, and mean for the three individual comparisons for all combinations of filters. The columns after that indicate the accuracy score for the average distance of the sum of MT and MV compared to TV, the difference between MT and MV compared to TV, and the ratio between the two groups.

5.2.3 Analysis

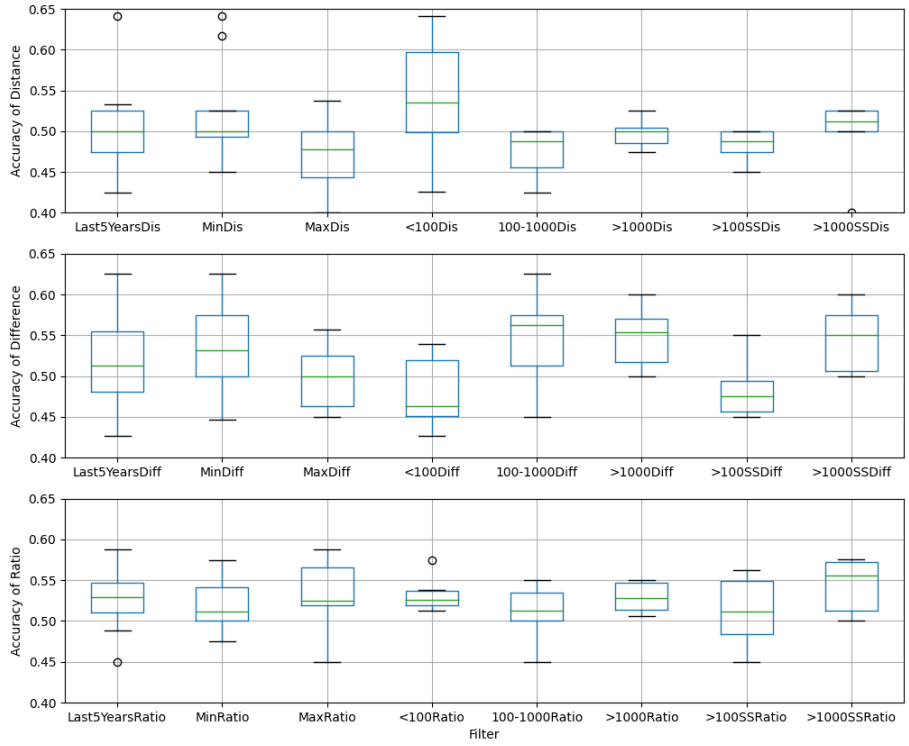

Figure 10: Boxplot of all accuracy scores of the three metrics (distance, difference, and ratio) for all filters that have been used more than once in combinations.

Figure 10 shows the boxplot for the additional three metrics for semantic change detection. The upper graph is the boxplot for the accuracy of the distance, the middle graph is the boxplot for the difference and the lower graph is the boxplot for the ratio. In the top graph, it is easy to see that the filter for < 100 has the largest spread between its results. The other filters of the amount category are more stable in this respect. Compared to this, the last five years and the min filter show noticeable outliers. In the lower graph, the differences between the individual results of the filters are not too large, while in the middle graph there are large differences between individual filters. For example, the values of the last five years for difference vary between 0.425 and 0.625. However, one can see that in general, the results are

nevertheless only on average 0.1 plus and minus away from the random baseline of 0.5. This means that the classification happens basically randomly. If only about half of all predictions are correct, one can assume that the models do not learn anything at all and therefore cannot make any statement about the decisions to classify a word.

In general, one would expect some sort of correlation between the token count and the resulting scores. Figure 11 shows the frequency counts of all words in the test set for each level of combinations. It is evident that the number of words in the test set that actually appear in all three subcorpora is a magnitude smaller for each level of combination. In more extreme cases, the words do not appear at all. There is a shift of the overall frequency profile to a lower range. One would assume that such a shift would lead to a decrease in performance. Judging by the results depicted in the aforementioned tables and the tables in the appendix, this doesn't seem to be the case.

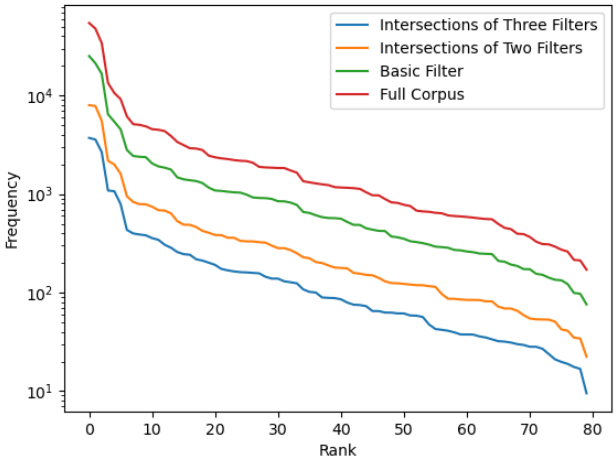

Figure 11: Frequency plot of all words appearing in the CanEn test set sorted by their rank for all different levels of filtering.

Figure 12 shows a scatterplot of the frequencies for the two different classes for all levels of filtering. The general trend is that the total counts for each word get lower the more filters are involved. The inherent distribution of the counts stays the same across the four different levels. The only thing that changes is the frequency of each word appearing and as Antoniak and Mimno (2018) pointed out, word embedding models tend to have unstable results when it comes to low-frequency counts of words, and as some of the words, especially in the bottom figure, have counts barely above 10, there is an inherent lack of credibility for the results.

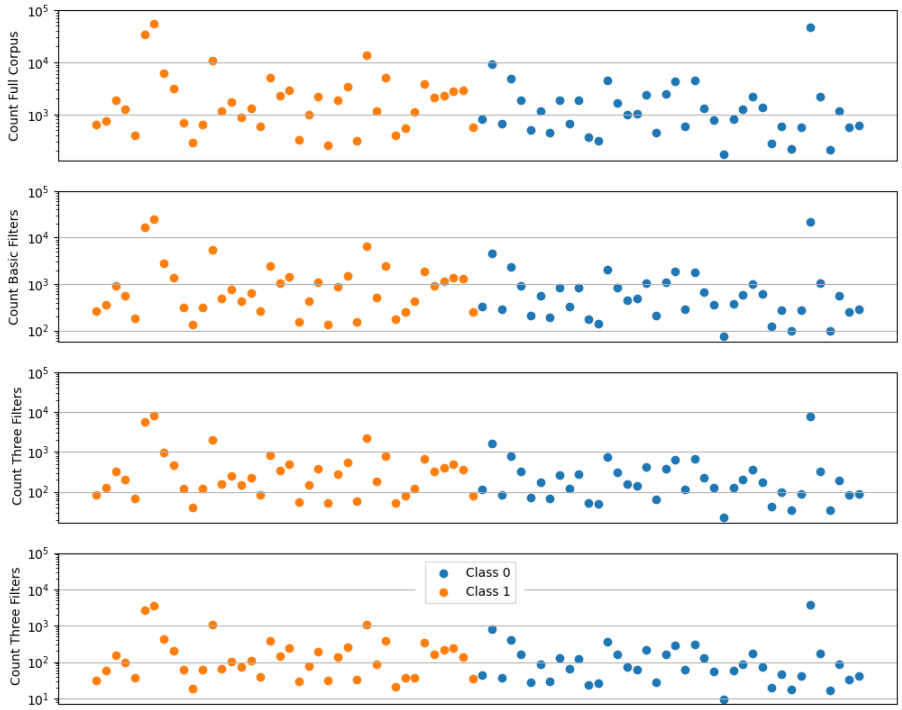

Figure 12: Scatterplot of the frequencies of the stable words (blue) and shifting words (orange) for all different levels of filtering.

Figure 13 shows a scatterplot for the additional three metrics for semantic change detection. Similarly to word similarity, I calculated the correlation between the accuracy scores and the token count of the models by calculating Spearman's correlation. In particular, for the metrics of difference ($\rho = 0.693$) and ratio ($\rho = 0.465$), a moderate to strong correlation is evident. However, only a weak correlation can be detected for the distance metric ($\rho = 0.294$). Especially the four models in blue at the very left of the figure support the argument that the results are arbitrary. Their token count is completely negligible compared to the other models, and yet extremely good results are achieved, especially in the upper graph by a subset of these models. As such findings cannot - at least to my knowledge - be found in this area of research, I would assume that the models' performances are more or less random as Miletic et al. (2021) achieve significantly better results on this task, dataset and Skip-gram architecture, with a mean accuracy (for the full corpus) of ≈ 0.700.

The results of the best possible splits shown in the appendix only reinforce this argument. For the majority of the models, the best split is not a balanced split approximately in the middle, but an extremely biased split in one direction, which only leads to the fact that the results per single direct comparison of the subcorpora are only minimally better than 0.5. Essentially, this means that the performance of the models depends, more or less, on chance.

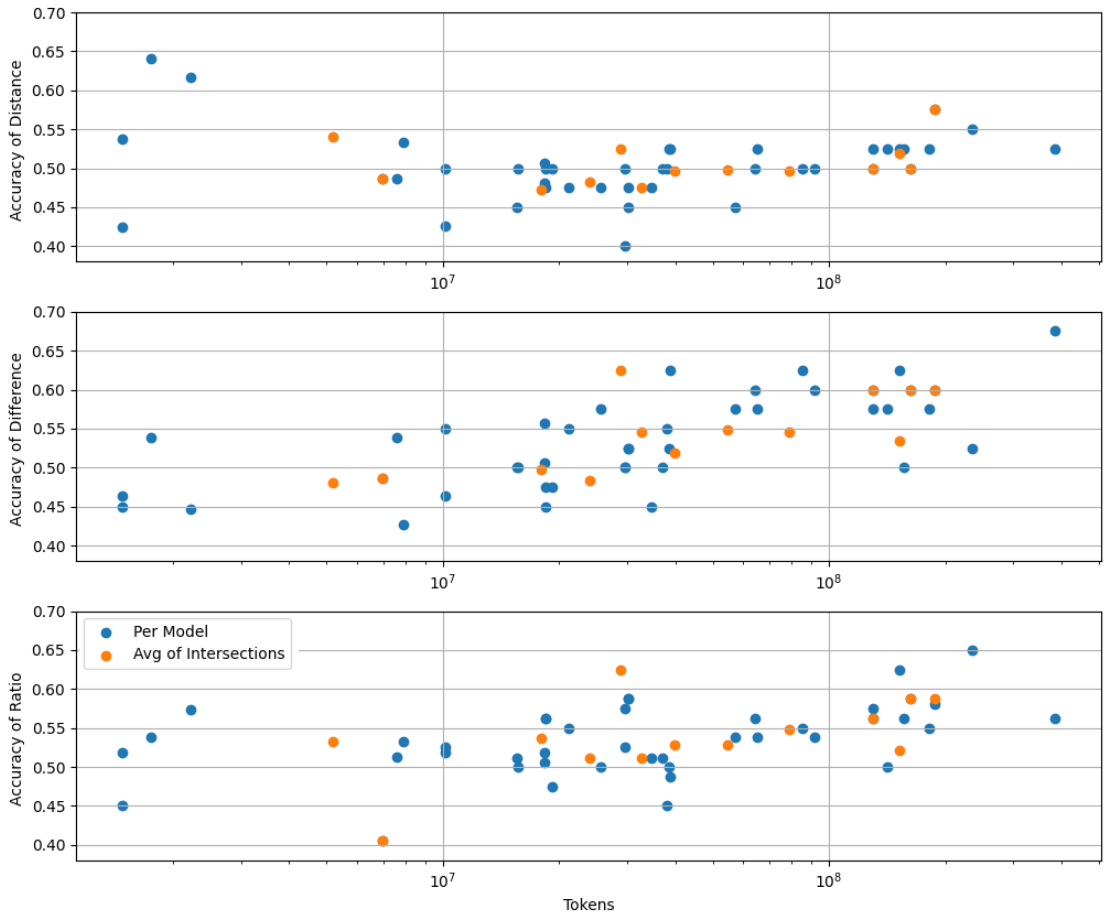

Figure 13: Scatterplot of the accuracy score and token count for the (i) average distance MT + MV , (ii) the difference of MT + MV compared to TV and (iii) the ratio between the average and the TV distance for each combination of filters (in blue) and averaged over all filters that appear in more than one combination (in orange).

5.3 Summary

This section presented the findings and analysis of the experiments related to word similarity and semantic change detection based on the previously defined filtering decisions. The experiments evaluated different filters applied to the same subcorpora and their impact on the performance of the respective tasks. The evaluation for semantic change detection proved to be difficult as the accuracy scores show similar performance across many filters, and the difference between the best and worst-performing filters is between ± 0.1 around the chance performance. Word similarity, in contrast, had clear-cut results. The combination of the findings of the two tasks yields the main findings of this thesis:

- Filters used in combination with other filters produce worse results on average than those used individually.
- There is a stark difference in performance between using the first and the last years as input due to large differences in the token count.
- The number of tweets per user and the length of a tweet have a large influence on the performance, with larger boundaries in the form of higher values for the limits leading to better results.
- The completely preprocessed but unfiltered corpus delivers the best average performance, but other models with significantly fewer tokens achieve comparable results.

To answer the hypotheses at the beginning of the thesis with these findings, it seems that the most common steps to limit the influence of highly-active individuals have already been accomplished by subsampling their tweets and setting a minimum length limit. However, these users do not seem to have a particularly high influence, since by limiting the number of tweets to a maximum of 1000 per user, only a marginally worse result is obtained compared to the original corpus. And since the corpus has already been cleaned up in terms of spam or near-duplicates, the word frequencies do not seem to have been significantly altered by these users.

6 Conclusion and Outlook

This thesis was concerned with the evaluation and investigation of filtering methods that aim to improve the distribution of tweets across users in a large Twitter corpus. The objective was to find out if there are filtering methods that provide a significant improvement of the original data without losing essential context. For this purpose, a complex pipeline consisting of preprocessing steps, filters and their combinations, and two subsequent downstream tasks in the form of word similarity and semantic change detection to evaluate the models were constructed to create a large-scale basis for comparison. Although a large number of filters and their combinations were investigated, no significant improvement over the original preprocessed corpus was found. It had an average correlation score of 0.623 for word similarity and the best possible accuracy values for almost all metrics for semantic change detection.

The models that rely on a subsampling technique, which limits the amount of tweets per user to 1000, and in case the user writes more than a thousand tweets, the tweets are randomly subsampled, had the strongest results for word similarity with an average correlation score of 0.601, and similarly strong results for the different accuracy metrics for semantic change detection even though this filter participated in a large number of combinations with other filters. The comparatively best results without combinations were achieved by the filter category length, which limited all tweets to a certain minimum length. Not only did the filters based on a tweet having 25%, 50% and 75% of the average tweet length achieve high values comparatively, but not necessarily better, than the results of the original corpus, but they even managed this feat with significantly fewer tokens and the associated shorter training time.

The logical explanation behind this could be the fact that the quality of Twitter data depends very much on its length, and excluding tweets below a certain threshold correlates strongly with a better result. However, to further support this claim, a large-scale analysis of all possible tweet lengths starting from the smallest length up to the maximum possible tweet length has to take place. To the extent that has

taken place here, one can indeed see a tendency towards a certain limit at which the performance of the models does not improve significantly despite a larger amount of data. However, no generalized statement regarding all Twitter corpora can be derived from this. Since a large number of research already limits the tweet length to a certain minimum size and subsampling tweets is the standard approach, the common practice seems to be already a reasonable solution. However, it should finally be said that the number of tokens does not necessarily lead to a better result. The fully preprocessed but unfiltered corpus achieved the best results, but other filters such as the filters of length as described above achieved comparable results with considerably fewer tokens in the magnitude of hundreds of millions.

6.1 Future Work

First of all, it should be mentioned that especially the already existing preprocessing steps are worthy of a closer examination. The bottom line is that every single one of these steps involves a change of the raw data and thus also a change of the resulting vectors. To what extent these changes in the original data have an effect on the resulting vectors should therefore be examined and analyzed on a large scale. Whether and to which extent the respective steps of the preprocessing justify an analysis of this scale is also worth considering as Saif et al. (2014) or Angiani et al. (2016) have investigated the effect of preprocessing steps in sentiment analysis and have come to the conclusion that every single step serves a purpose in improving the quality of data.

Secondly, the filtering process itself should be applied even more extensively. The division into the two extremes as it is for the length or the year of the tweets should be extended by an iterative filtering process. Thus, the respective length filters should also be included in the combinations which would have been part of the thesis if not for the large extension of the models to be analyzed which was not feasible in the given time frame. If necessary, the concrete values of the filters would still have to be adjusted in order to guarantee a broad analysis.

Possible extensions in this template pipeline include on the one hand the inclusion of further preprocessing steps like part of speech tagging, the identification of the language, and the replacement of misspelled words by the correct spelling. On the other hand, the filtering process itself can be extended to include several categories so that the large-scale investigation is not limited to the three categories selected here. A possible field now would be the division into genders, age groups, day/ night activities, or the percentage of tweets in a foreign language. It would also be interesting to examine the type of tweets themselves, whether they are tweets or mentions, retweets, or quoted tweets.

Regardless of what serves as input, a change in the training architecture may also be considered. Word2Vec offers two different context representations (CBOW and Skip-gram) with different optimization objectives (e.g. the used negative sampling). Changing these parameters and other hyperparameters like the dimensionality of the vectors could lead to different results. Due to the stochastic nature of the training process, word embeddings can lead to surprisingly strong instabilities. A common compromise to counteract outliers and to guarantee stability is to run the same training of the word embeddings multiple times and then combine the results using the average (Wendlandt et al., 2018). Furthermore, there has been work done proposing a suitable feature selection for word2vec model. (Tian et al., 2018) This is another sanity check for the general validity of the work of this thesis. Furthermore, there are other, token-based or type-based architectures for embedding the meaning of words such as GloVe (Pennington et al., 2014) or BERT (Devlin et al., 2019), which might provide at least competitive results compared to SGNS (Laicher et al., 2021; Jain, 2020). The experimental setup for BERT, due to being already pretrained, is altered in such a way, that fine-tuning the architecture with each resulting corpus after filtering decisions, is then the model to be evaluated.

Another extension could be the addition of multiple test sets. Particularly for word similarity, there are still further evaluation options in the form of further intrinsic test sets or even the extension to extrinsic surveys. Because the test set for semantic change detection was specifically created for that data set, and similar test sets were similarly created for a specific data set, language, or domain, extension to other test sets for this task is difficult. It seems logical to create another test set specially created for this data set, but whether this is the right approach does not seem conclusive, at least to me. The problem, that the models have too few tokens as input data remains and the search for relevant words of French origin or use that can be found more or less randomly in these small models seems very contrived and it boils down to just having more of the same of the current evaluation method.

In addition, a look should be taken at other possible downstream tasks. Many (if not all) tasks in the area of lexical semantics use word vectors to embed the meaning of words and should therefore be reasonably applicable here. The inclusion of further tasks guarantees an even broader and more coherent analysis and, if applicable, consolidates the findings in this thesis. In conclusion, the scheme of all methods and evaluation tasks described here in detail can be easily applied to another, new user-generated Twitter corpus.

Finally, it should be said that even the most sophisticated filtering and preprocessing methods fail if the number of resulting tokens is too low and the results word vectors are too unreliable to allow any coherent analysis that isn't based on inherent random behaviour. Nevertheless, it is important to say that the finding of a certain limit of tokens, above which the performance does not necessarily improve, is not limited to this thesis, but is already known in many fields and areas. For example, Lin et al. (2022) or Kolisetty and Rajput (2019) report about this phenomenon that more training data does not necessarily lead to better results in different fields of machine learning. To what extent this holds true for all tasks, not only in the field of natural language processing relying on word embeddings, but for all tasks in machine learning, and whether all corpora have this innate ability is left to be determined.

References

Agirre, E., Alfonseca, E., Hall, K., Kravalova, J., Pasca, M., and Soroa, A. (2009). A Study on Similarity and Relatedness Using Distributional and WordNet-based Approaches. In *Proceedings of the 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL-2009)*, pages 19–27, Boulder, Colorado.

Al Sharou, K., Li, Z., and Specia, L. (2021). Towards a Better Understanding of Noise in Natural Language Processing. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP 2021)*, pages 53–62, Held Online. INCOMA Ltd.

Angiani, G., Ferrari, L., Fontanini, T., Fornacciari, P., Iotti, E., Magliani, F., and Manicardi, S. (2016). A Comparison Between Preprocessing Techniques for Sentiment Analysis in Twitter. In *International Workshop on Knowledge Discovery on the Web*.

Antoniak, M. and Mimno, D. (2018). Evaluating the Stability of Embedding-based Word Similarities. *Transactions of the Association for Computational Linguistics*, 6:107–119.

Artetxe, M., Labaka, G., Lopez-Gazpio, I., and Agirre, E. (2018). Uncovering Divergent Linguistic Information in Word Embeddings with Lessons for Intrinsic and Extrinsic Evaluation. In *Proceedings of the 22nd Conference on Computational Natural Language Learning*, pages 282–291, Brussels, Belgium. Association for Computational Linguistics.

Babić, K., Martinčić-Ipšić, S., and Meštrović, A. (2020). Survey of Neural Text Representation Models. *Information*, 11(11).

Bao, Y., Quan, C., Wang, L., and Ren, F. (2014). The Role of Preprocessing in Twitter Sentiment Analysis. In Huang, D.-S., Jo, K.-H., and Wang, L., editors, *In-*

telligent Computing Methodologies, pages 615–624, Cham. Springer International Publishing.

Bian, J., Liu, Y., Agichtein, E., and Zha, H. (2008). Finding the Right Facts in the Crowd: Factoid Question Answering over Social Media. In *Proceedings of the 17th International Conference on World Wide Web*, WWW '08, page 467–476, New York, NY, USA. Association for Computing Machinery.

Bird, S., Klein, E., and Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit.* " O'Reilly Media, Inc.".

Boot, A., Sang, E., Dijkstra, K., and Zwaan, R. (2019). How Character Limit Affects Language Usage in Tweets. *Palgrave Communications*, 5.

Bruni, E., Tran, N. K., and Baroni, M. (2014). Multimodal Distributional Semantics. *Journal of Artificial Intelligence Research.*

Chai, C. P. (2023). Comparison of Text Preprocessing Methods. *Natural Language Engineering*, 29(3):509–553.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., and Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407.

Devlin, J., Chang, M., Lee, K., and Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186. Association for Computational Linguistics.

Doval, Y., Vilares, J., and Gómez-Rodríguez, C. (2020). Towards robust word embeddings for noisy texts. *Applied Sciences*, 10:6893.

El-Haj, M., Kruschwitz, U., and Fox, C. (2010). Using Mechanical Turk to Create a Corpus of Arabic Summaries. In *Proceedings of the 7th International Conference on Language Resources and Evaluation : Workshops & Tutorials May 17-18, May 22-23, Main Conference May 19-21, Valletta*. ELRA, Paris.

Elekes, A., Schaeler, M., and Boehm, K. (2017). On the Various Semantics of Similarity in Word Embedding Models. Technical Report 3, Karlsruher Institut für Technologie (KIT).

Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

Faruqui, M., Tsvetkov, Y., Rastogi, P., and Dyer, C. (2016). Problems With Evaluation of Word Embeddings Using Word Similarity Tasks. In *Proceedings of the 1st Workshop on Evaluating Vector-Space Representations for NLP*, pages 30–35, Berlin, Germany. Association for Computational Linguistics.

Finkelstein, L., Gabrilovich, E., Matias, Y., Rivlin, E., Solan, Z., Wolfman, G., and Ruppin, E. (2001). Placing search in context: The concept revisited. *ACM Transactions on Information Systems - TOIS*, 20:406–414.

Firth, J. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. reprinted in Palmer, F. (ed. 1968) Selected Papers of J. R. Firth, Longman, Harlow.

Freedman, D., Pisani, R., and Purves, R. (2007). Statistics (international student edition). *Pisani, R. Purves, 4th edn. WW Norton & Company, New York*.

Gennaro, G., Buonanno, A., and Palmieri, F. (2021). Considerations about learning word2vec. *The Journal of Supercomputing*, 77.

Gligorić, K., Anderson, A., and West, R. (2018). How Constraints Affect Content: The Case of Twitter's Switch from 140 to 280 Characters. *Proceedings of the International AAAI Conference on Web and Social Media*, 12(1).

Goldberg, Y. and Levy, O. (2014). word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722.

González, M. (2015). An Analysis of Twitter Corpora and the Differences between Formal and Colloquial Tweets. In *TweetMT@SEPLN*.

Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016). Diachronic Word Embeddings Reveal Statistical Laws of Semantic Change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Harris, Z. (1954). Distributional structure. *Word*, 10(23):146–162.

Hill, F., Reichart, R., and Korhonen, A. (2015). SimLex-999: Evaluating Semantic Models With (Genuine) Similarity Estimation. *Computational Linguistics*, 41(4):665–695.

Jain, V. (2020). GloVeInit at SemEval-2020 task 1: Using GloVe vector initialization for unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 208–213, Barcelona (online). International Committee for Computational Linguistics.

Kabakus, A. T. and Kara, R. (2017). A Survey of Spam Detection Methods on Twitter. *International Journal of Advanced Computer Science and Applications*, 8.

Kaiser, J., Kurtyigit, S., Kotchourko, S., and Schlechtweg, D. (2021). Effects of pre- and post-processing on type-based embeddings in lexical semantic change detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 125–137, Online. Association for Computational Linguistics.

Kaiser, J., Schlechtweg, D., and im Walde, S. S. (2020). OP-IMS @ DIACR-Ita: Back to the Roots: SGNS+OP+CD still rocks Semantic Change Detection. *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*.

Klein, A. Z., Magge, A., O'Connor, K., Flores Amaro, J. I., Weissenbacher, D., and Gonzalez Hernandez, G. (2021). Toward Using Twitter for Tracking COVID-19: A Natural Language Processing Pipeline and Exploratory Data Set. *J Med Internet Res*, 23(1):e25314.

Kolisetty, V. and Rajput, D. (2019). A Review on the Significance of Machine Learning for Data Analysis in Big Data. *Jordanian Journal of Computers and Information Technology*, 06:1.

Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. (2018). Diachronic word embeddings and semantic shifts: a survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Laicher, S., Baldissin, G., Castañeda, E., Schlechtweg, D., and Schulte Im Walde, S. (2020). CL-IMS @ DIACR-Ita: Volente o Nolente: BERT does not outperform SGNS on Semantic Change Detection. *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*.

Laicher, S., Kurtyigit, S., Schlechtweg, D., Kuhn, J., and Schulte im Walde, S. (2021). Explaining and improving BERT performance on lexical semantic change detection. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 192–202, Online. Association for Computational Linguistics.

Li, Q., Shah, S., Liu, X., and Nourbakhsh, A. (2017). Data Sets: Word Embeddings Learned from Tweets and General Data. *Proceedings of the International AAAI Conference on Web and Social Media*, 11.

Lin, J., Zhang, A., Lecuyer, M., Li, J., Panda, A., and Sen, S. (2022). Measuring the effect of training data on deep learning predictions via randomized experiments. In *International Conference on Machine Learning*.

Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient Estimation of Word Representations in Vector Space. In Bengio, Y. and LeCun, Y., editors, *1st*

International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed Representations of Words and Phrases and their Compositionality. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

Miletic, F., Przewozny-Desriaux, A., and Tanguy, L. (2020). Collecting Tweets to Investigate Regional Variation in Canadian English. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 6255–6264, Marseille, France. European Language Resources Association.

Miletic, F., Przewozny-Desriaux, A., and Tanguy, L. (2021). Detecting Contact-Induced Semantic Shifts: What Can Embedding-Based Methods Do in Practice? In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10852–10865, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Moreno-Ortiz, A. and García-Gámez, M. (2023). Strategies for the Analysis of Large Social Media Corpora: Sampling and Keyword Extraction Methods. *Corpus Pragmatics*, 7:1–25.

Myers, J., Well, A., and Lorch Jr, R. (2010). *Research Design and Statistical Analysis: Third Edition (3rd ed.)*. Routledge.

Naseem, U., Razzak, I., Khan, S. K., and Prasad, M. (2020). A Comprehensive Survey on Word Representation Models: From Classical to State-of-the-Art Word Representation Language Models. *Transactions on Asian and Low-Resource Language Information Processing*, 20:1 – 35.

Navigli, R. and Martelli, F. (2019). An Overview of Word and Sense Similarity. *Natural Language Engineering*, 25(6):693–714.

Neubig, G. and Duh, K. (2013). How Much Is Said in a Tweet? A Multilingual, Information-theoretic Perspective. In *AAAI Spring Symposium: Analyzing Microtext*.

Oberbichler, S. and Pfanzelter, E. (2021). Topic-specific corpus building: A step towards a representative newspaper corpus on the topic of return migration using text mining methods. *Journal of Digital History*, 1(1):74–98.

Osgood, C. E. (1964). Semantic differential technique in the comparative study of cultures. *American Anthropologist*, 66(3):171–200.

Pak, A. and Paroubek, P. (2010). Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Palomino, M. A. and Aider, F. (2022). Evaluating the Effectiveness of Text Pre-Processing in Sentiment Analysis. *Applied Sciences*, 12(17).

Pennington, J., Socher, R., and Manning, C. (2014). GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Ramachandran, D. and Parvathi, R. (2019). Analysis of Twitter Specific Preprocessing Technique for Tweets. *Procedia Computer Science*, 165:245–251.

Rehurek, R. and Sojka, P. (2011). Gensim–Python Framework for Vector Space Modelling. *NLP Centre, Faculty of Informatics, Masaryk University, Brno, Czech Republic*, 3(2).

Şahinuç, F. and Toraman, C. (2021). Tweet Length Matters: A Comparative Analysis on Topic Detection in Microblogs. In Hiemstra, D., Moens, M.-F., Mothe, J., Perego, R., Potthast, M., and Sebastiani, F., editors, *Advances in Information Retrieval*, pages 471–478, Cham. Springer International Publishing.

Sahlgren, M. (2006). *The Word-Space Model: Using Distributional Analysis to Represent Syntagmatic and Paradigmatic Relations between Words in High-Dimensional Vector Spaces*. PhD thesis, Stockholm University, Stockholm, Sweden.

Saif, H., Fernandez, M., He, Y., and Alani, H. (2014). On Stopwords, Filtering and Data Sparsity for Sentiment Analysis of Twitter. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 810–817, Reykjavik, Iceland. European Language Resources Association (ELRA).

Sastry, N. (2012). How To Tell Head From Tail in User-generated Content Corpora. *Proceedings of the International AAAI Conference on Web and Social Media*, 6.

Schlechtweg, D., Hätty, A., Del Tredici, M., and Schulte im Walde, S. (2019). A Wind of Change: Detecting and Evaluating Lexical Semantic Change across Times and Domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy. Association for Computational Linguistics.

Schlechtweg, D., McGillivray, B., Hengchen, S., Dubossarsky, H., and Tahmasebi, N. (2020). SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Schütze, H. (1993). A vector model for Syntagmatic and Paradigmatic Relatedness. In *Making sense of words*, pages 104–113. Ninth Annual Conference of the UW Centre for the New OED and Text Research.

Shoemark, P., Liza, F. F., Nguyen, D., Hale, S., and McGillivray, B. (2019). Room to Glo: A Systematic Comparison of Semantic Change Detection Approaches with Word Embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 66–76, Hong Kong, China. Association for Computational Linguistics.

Symeonidis, S., Effrosynidis, D., and Arampatzis, A. (2018). A Comparative Evaluation of Preprocessing Techniques and their Interactions for Twitter Sentiment Analysis. *Expert Systems with Applications*, 110:298–310.

Tahmasebi, N., Borin, L., Jatowt, A., Xu, Y., and Hengchen, S. (2021). *Computational Approaches to Semantic Change*. Language Science Press, Berlin.

Taie, M., Kadry, S., and Lucas, J. (2019). Online Data Preprocessing: A Case Study Approach. *International Journal of Electrical and Computer Engineering*, 9:2620–2626.

Tan, L., Zhang, H., Clarke, C., and Smucker, M. (2015). Lexical Comparison Between Wikipedia and Twitter Corpora by Using Word Embeddings. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 657–661, Beijing, China. Association for Computational Linguistics.

Tian, W., Li, J., and Li, H. (2018). A Method of Feature Selection Based on Word2Vec in Text Categorization. In *2018 37th Chinese Control Conference (CCC)*, pages 9452–9455.

Wang, B., Wang, A., Chen, F., Wang, Y., and Kuo, C.-C. J. (2019). Evaluating Word Embedding Models: Methods and Experimental Results. *APSIPA Transactions on Signal and Information Processing*, 8(1).

Wendlandt, L., Kummerfeld, J. K., and Mihalcea, R. (2018). Factors Influencing the Surprising Instability of Word Embeddings. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2092–2102, New Orleans, Louisiana. Association for Computational Linguistics.

Wes McKinney (2010). Data Structures for Statistical Computing in Python. In *Proceedings of the 9th Python in Science Conference*, pages 56– 61.

Zhao, Y., Yin, P., Li, Y., He, X., Du, J., Tao, C., Guo, Y., Prosperi, M., Veltri, P., Yang, X., Wu, Y., and Bian, J. (2022). Data and Model Biases in Social Media Analyses: A Case Study of COVID-19 Tweets. *AMIA Annual Symposium Proceedings*, 2021:1264–1273.

All links were last followed on the 14th of August 2023.

7 Appendix

7.1 Complete Tables for Word Similarity

Raw and Full Corpus

Corpus - Test Set	Raw Data	Full Data
Montreal - MEN	0.650/0.651/0.0	0.739/0.746/4.5
Montreal - SimLex999	0.258/0.250/0.3	0.330/0.314/0.0
Montreal - 353 Similarity	0.664/0.656/0.5	0.754/0.760/0.0
Montreal - 353 Related	0.584/0.585/0.4	0.646/0.664/0.0
avg	0.539/0.536	0.617/0.621
Toronto - MEN	0.668/0.672/0.1	0.746/0.751/2.9
Toronto - SimLex999	0.255/0.242/0.2	0.346/0.328/0.0
Toronto - 353 Similarity	0.649/0.645/0.5	0.748/0.751/0.0
Toronto - 353 Related	0.600/0.584/0.4	0.657/0.674/0.0
avg	0.543/0.536	0.624/0.625
Vancouver - MEN	0.672/0.674/0.1	0.759/0.764/3.6
Vancouver - SimLex999	0.264/0.251/0.1	0.346/0.329/0.0
Vancouver - 353 Similarity	0.658/0.644/1.0	0.757/0.758/0.0
Vancouver - 353 Related	0.613/0.614/1.2	0.649/0.663/0.0
avg	0.552/0.546	0.628/0.629
avg total	0.545/0.539	0.623/0.625

Table 21: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set using the raw and the full corpus.

Filtering By Amount

	Amount of Tweets				
Corpus - Test Set	<100	100-1000	>1000	>100SS	>1000SS
M - MEN	0.674/0.681/16.4	0.729/0.735/6.2	0.739/0.741/5.8	0.712/0.721/9.5	0.737/0.734/4.0
M - SimLex999	0.248/0.244/0.4	0.321/0.304/0.0	0.319/0.302/0.0	0.305/0.294/0.0	0.335/0.320/0.0
M - 353 Sim	0.674/0.685/3.4	0.737/0.741/0.0	0.749/0.750/0.0	0.721/0.724/0.0	0.744/0.747/0.0
M - 353 Related	0.621/0.634/2.8	0.635/0.648/0.4	0.654/0.667/0.0	0.637/0.663/0.8	0.648/0.664/0.0
average	0.554/0.561	0.605/0.607	0.615/0.615	0.594/0.601	0.616/0.616
T - MEN	0.674/0.684/16.0	0.733/0.739/5.1	0.736/0.742/3.2	0.729/0.739/9.1	0.746/0.752/3.0
T - SimLex999	0.225/0.225/0.2	0.324/0.306/0.0	0.323/0.313/0.0	0.305/0.293/0.1	0.334/0.317/0.0
T - 353 Sim	0.661/0.665/2.5	0.746/0.751/0.0	0.741/0.745/0.0	0.731/0.732/0.5	0.736/0.737/0.0
T - 353 Related	0.622/0.633/4.0	0.648/0.674/0.4	0.659/0.677/0.0	0.649/0.664/0.4	0.636/0.652/0.0
average	0.546/0.552	0.613/0.618	0.615/0.619	0.604/0.607	0.613/0.615
V - MEN	0.676/0.686/16.4	0.741/0.747/5.7	0.748/0.753/4.5	0.734/0.741/9.9	0.748/0.752/3.4
V - SimLex999	0.249/0.239/0.5	0.331/0.314/0.0	0.339/0.322/0.0	0.315/0.298/0.0	0.338/0.320/0.0
V - 353 Sim	0.697/0.708/2.5	0.750/0.754/0.0	0.753/0.748/0.0	0.744/0.749/0.0	0.751/0.749/0.0
V - 353 Related	0.638/0.657/3.6	0.645/0.659/0.4	0.650/0.664/0.0	0.673/0.684/0.4	0.636/0.657.0.0
average	0.565/0.573	0.617/0.619	0.622/0.622	0.617/0.618	0.618/0.620
average total	0.555/0.562	0.612/0.614	0.620/0.619	0.605/0.608	0.617/0.617

Table 22: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the amount of tweets per user.

Filtering By Length

	Length of Tweet					
Corpus - Test Set	25	50	75	min	max	mean
M - MEN	0.741/0.748/5.1	0.738/0.744/5.1	0.735/0.741/5.2	0.712/0.723/8.4	0.717/0.724/11.6	0.736/0.743/5.1
M - SimLex999	0.330/0.313/0.0	0.320/0.302/0.0	0.331/0.317/0.0	0.299/0.292/0.0	0.317/0.305/0.2	0.321/0.306/0.0
M - 353 Sim	0.738/0.737/0.0	0.738/0.734/0.0	0.756/0.755/0.0	0.706/0.710/0.0	0.728/0.729/0.0	0.745/0.744/0.0
M - 353 Related	0.638/0.653/0.0	0.637/0.658/0.0	0.656/0.678/0.0	0.627/0.641/0.0	0.650/0.669/0.4	0.647/0.668/0.0
avg	0.612/0.613	0.608/0.610	0.612/0.623	0.586/0.592	0.603/0.607	0.612/0.615
T - MEN	0.748/0.753/3.0	0.751/0.756/3.3	0.749/0.754/4.3	0.716/0.723/6.5	0.732/0.736/11.6	0.747/0.751/4.1
T - SimLex999	0.340/0.318/0.0	0.340/0.320/0.0	0.342/0.322/0.0	0.289/0.280/0.0	0.319/0.304/0.1	0.333/0.314/0.0
T - 353 Sim	0.756/0.761/0.0	0.734/0.740/0.0	0.739/0.741/0.0	0.731/0.730/0.0	0.702/0.703/0.0	0.737/0.738/0.0
T - 353 Related	0.651/0.671/0.0	0.632/0.655/0.0	0.642/0.661/0.0	0.627/0.640/0.4	0.611/0.620/0.4	0.632/0.657/0.0
avg	0.624/0.626	0.614/0.612	0.618/0.620	0.590/0.593	0.591/0.591	0.612/0.615
V - MEN	0.751/0.756/3.6	0.749/0.753/3.6	0.750/0.754/4.5	0.724/0.732/6.8	0.721/0.724/9.8	0.753/0.757/4.3
V - SimLex999	0.333/0.312/0.0	0.332/0.312/0.0	0.333/0.314/0.0	0.316/0.305/0.0	0.311/0.295/0.0	0.334/0.315/0.0
V - 353 Sim	0.773/0.771/0.0	0.771/0.768/0.0	0.771/0.773/0.0	0.713/0.708/0.5	0.748/0.748/0.5	0.775/0.776/0.0
V - 353 Related	0.651/0.667/0.0	0.647/0.664/0.0	0.653/0.675/0.0	0.633/0.643/0.0	0.658/0.681/0.4	0.645/0.664/0.0
avg	0.627/0.627	0.625/0.625	0.627/0.630	0.596/0.598	0.610/0.612	0.627/0.628
avg total	0.621/0.622	0.616/0.617	0.621/0.624	0.591/0.594	0.601/0.603	0.617/0.619

Table 23: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the length of the tweets.

Filtering By Time

	Time of Tweet	
Corpus - Test Set	First 5	Last 5
Montreal - MEN	0.638/0.648/17.5	0.726/0.735/5.2
Montreal - SimLex999	0.174/0.177/0.6	0.328/0.306/0
Montreal - 353 Similarity	0.657/0.674/5.9	0.744/0.745/0
Montreal - 353 Related	0.609/0.619/7.5	0.644/0.657/0
avg	0.520/0.530	0.610/0.611
Toronto - MEN	0.662/0.672/14.6	0.741/0.744/3.3
Toronto - SimLex999	0.213/0.207/0.3	0.336/0.317/0
Toronto - 353 Similarity	0.666/0.674/3.9	0.711/0.712/0
Toronto - 353 Related	0.637/0.656/5.6	0.623/0.644/0
avg	0.545/0.552	0.603/0.604
Vancouver - MEN	0.677/0.688/13.7	0.754/0.757/5.0
Vancouver - SimLex999	0.214/0.219/0.2	0.343/0.325/0.0
Vancouver - 353 Similarity	0.674/0.672/3.0	0.752/0.754/0.0
Vancouver - 353 Related	0.608/0.630/3.6	0.640/0.658/0.0
avg	0.543/0.552	0.622/0.624
avg total	0.536/0.545	0.612/0.613

Table 24: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the time (year) of the tweets.

Filtering By Amount and Length

	... Amount of Tweets and Minimum Tweet Length				
Corpus - Test Set	<100	100-1000	>1000	>100SS	>1000SS
M - MEN	0.573/0.576/23.3	0.703/0.715/10.2	0.684/0.693/10.0	0.683/0.692/11.1	0.709/0.721/8.4
M - SimLex999	0.183/0.186/4.6	0.250/0.244/0.0	0.272/0.268/0.0	0.242/0.238/0.1	0.296/0.285/0.0
M - 353 Sim	0.592/0.592/8.4	0.713/0.729/0.0	0.702/0.708/1.0	0.691/0.699/1.0	0.713/0.726/0.0
M - 353 Related	0.502/0.508/8.3	0.636/0.638/1.2	0.627/0.637/1.2	0.598/0.601/1.2	0.611/0.637/0.8
average	0.462/0.466	0.576/0.582	0.571/0.576	0.554/0.558	0.582/0.592
T - MEN	0.554/0.557/24.5	0.689/0.699/10.3	0.700/0.705/7.2	0.695/0.706/11.1	0.712/0.718/6.5
T - SimLex999	0.163/0.154/5.1	0.240/0.233/0.1	0.279/0.272/0.0	0.246/0.244/0.1	0.291/0.282/0.0
T - 353 Sim	0.541/0.540/11.3	0.703/0.708/0.5	0.720/0.732/0.0	0.709/0.714/0.0	0.726/0.733/0.0
T - 353 Related	0.487/0.495/12.3	0.619/0.631/1.6	0.647/0.651/0.4	0.631/0.639/0.8	0.635/0.648/0.4
average	0.436/0.437	0.563/0.568	0.587/0.590	0.570/0.576	0.591/0.595
V - MEN	0.562/0.569/25.3	0.711/0.722/10.1	0.708/0.715/8.5	0.694/0.704/11.3	0.727/0.733/6.8
V - SimLex999	0.150/0.150/6.2	0.268/0.266/0.1	0.286/0.279/0.0	0.277/0.271/0.1	0.312/0.304/0.0
V - 353 Sim	0.574/0.562/12.3	0.704/0.709/1.5	0.703/0.704/0.5	0.703/0.702/1.0	0.706/0.706/0.5
V - 353 Related	0.539/0.543/10.7	0.649/0.663/2.4	0.656/0.676/0.8	0.621/0.630/2.0	0.639/0.652/0.0
average	0.456/0.456	0.583/0.590	0.588/0.594	0.574/0.577	0.596/0.599
average total	0.452/0.453	0.574/0.580	0.582/0.587	0.566/0.570	0.590/0.595

Table 25: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the amount and the length of the tweets. (1)

	... Amount of Tweets and Maximum Tweet Length				
Corpus - Test Set	<100	100-1000	>1000	>100SS	>1000SS
M - MEN	0.603/0.608/31.7	0.680/0.687/17.2	0.688/0.697/14.9	0.698/0.707/13.8	0.711/0.719/11.6
M - SimLex999	0.195/0.193/6.5	0.259/0.255/0.5	0.269/0.2596/0.2	0.300/0.293/0.3	0.313/0.304/0.2
M - 353 Sim	0.605/0.609/11.8	0.683/0.698/1.0	0.731/0.739/1.0	0.702/0.713/0.5	0.736/0.740/0.0
M - 353 Related	0.504/0.510/10.3	0.629/0.643/1.2	0.643/0.660/2.0	0.630/0.647/1.2	0.636/0.649/0.4
average	0.477/0.480	0.563/0.571	0.583/0.589	0.583/0.590	0.599/0.603
T - MEN	0.598/0.602/30.9	0.698/0.704/15.6	0.712/0.720/13.5	0.720/0.725/13.0	0.734/0.734/11.6
T - SimLex999	0.188/0.186/5.8	0.268/0.257/0.4	0.295/0.284/0.2	0.294/0.280/0.2	0.309/0.294/0.1
T - 353 Sim	0.605/0.599/14.3	0.716/0.724/2.0	0.676/0.675/1.0	0.687/0.681/1.0	0.708/0.708/0.0
T - 353 Related	0.523/0.521/11.5	0.614/0.629/2.4	0.612/0.622/1.6	0.605/0.610/1.6	0.645/0.655/0.4
average	0.479/0.477	0.574/0.579	0.574/0.575	0.577/0.574	0.599/0.598
V - MEN	0.611/0.618/30.4	0.685/0.691/14.0	0.718/0.724/12.7	0.716/0.721/12.2	0.721/0.725/9.8
V - SimLex999	0.141/0.147/5.5	0.287/0.279/0.4	0.292/0.279/0.0	0.302/0.289/0.0	0.321/0.308/0.8
V - 353 Sim	0.646/0.638/12.8	0.698/0.703/3.0	0.718/0.716/1.0	0.730/0.741/1.5	0.747/0.745/0.5
V - 353 Related	0.523/0.539/10.7	0.638/0.647/3.6	0.641/0.658/1.2	0.647/0.669/1.2	0.651/0.665/0.4
average	0.480/0.486	0.577/0.580	0.592/0.594	0.599/0.605	0.610/0.611
average total	0.479/0.481	0.571/0.576	0.583/0.586	0.586/0.590	0.603/0.604

Table 26: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the amount and the length of the tweets. (2)

Filtering By Amount and Time

	Last 5 Years and ... Amount of Tweets				
Corpus - Test Set	5 + <100	5+100-1000	5+>1000	5+>100SS	5+>1000SS
M - MEN	0.659/0.667/17.6	0.723/0.730/8.3	0.713/0.719/7.5	0.720/0.727/11.1	0.728/0.734/5.9
M - SimLex999	0.247/0.246/0.6	0.306/0.292/0.0	0.313/0.301/0.0	0.297/0.291/0.0	0.326/0.311/0.0
M - 353 Sim	0.656/0.665/3.4	0.724/0.730/0.0	0.748/0.750/0.0	0.699/0.701/0.0	0.730/0.731/0.0
M - 353 Related	0.600/0.620/3.2	0.632/0.648/0.4	0.646/0.665/0.0	0.608/0.630/0.4	0.626/0.646/0.4
average	0.541/0.550	0.596/0.600	0.605/0.609	0.581/0.587	0.603/0.605
T - MEN	0.659/0.668/16.6	0.738/0.745/7.3	0.732/0.736/5.7	0.728/0.736/8.8	0.746/0.752/4.1
T - SimLex999	0.239/0.235/0.2	0.314/0.299/0.0	0.325/0.317/0.0	0.306/0.291/0.0	0.336/0.318/0.0
T - 353 Sim	0.658/0.661/3.0	0.710/0.719/0.0	0.724/0.731/0.0	0.714/0.708/0.5	0.714/0.712/0.0
T - 353 Related	0.606/0.621/4.4	0.649/0.665/0.4	0.646/0.661/0.4	0.626/0.628/1.6	0.630/0.647/0.0
average	0.541/0.546	0.603/0.607	0.607/0.611	0.594/0.591	0.607/0.607
V - MEN	0.674/0.682/16.5	0.747/0.749/7.8	0.751/0.756/6.8	0.723/0.729/8.5	0.752/0.756/5.8
V - SimLex999	0.237/0.225/0.7	0.323/0.306/0.0	0.329/0.314/0.0	0.304/0.293/0.0	0.349/0.332/0.0
V - 353 Sim	0.678/0.679/5.4	0.754/0.761/0.0	0.755/0.754/0.0	0.741/0.749/0.0	0.756/0.753/0.0
V - 353 Related	0.604/0.604/5.2	0.643/0.650/0.4	0.664/0.686/0.0	0.654/0.677/0.8	0.650/0.663/0.0
average	0.548/0.550	0.617/0.616	0.625/0.628	0.605/0.612	0.627/0.626
average total	0.544/0.548	0.605/0.608	0.612/0.616	0.593/0.597	0.612/0.613

Table 27: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the amount and the time of the tweets.

Filtering By Length and Time

	Last 5 Years and Min/Max Tweet Length	
Corpus - Test Set	Last 5 and Min Length	Last 5 and Max Length
M - MEN	0.697/0.708/10.4	0.712/0.719/11.6
M - SimLex999	0.277/0.271/0.0	0.305/0.298/0.2
M - 353 Sim	0.694/0.705/0.0	0.729/0.733/0.0
M - 353 Related	0.629/0.645/1.2	0.618/0.637/0.4
average	0.574/0.582	0.589/0.597
T - MEN	0.692/0.699/9.5	0.731/0.736/11.6
T - SimLex999	0.275/0.270/0.0	0.316/0.299/0.1
T - 353 Sim	0.717/0.732/0.0	0.707/0.705/0.0
T - 353 Related	0.638/0.656/0.4	0.630/0.645/0.4
average	0.581/0.589	0.596/0.595
V - MEN	0.716/0.727/8.7	0.723/0.727/9.8
V - SimLex999	0.281/0.276/0.0	0.315/0.298/0.0
V - 353 Sim	0.713/0.708/1.0	0.760/0.759/0.5
V - 353 Related	0.642/0.660/0.8	0.662/0.672/0.4
average	0.588/0.593	0.615/0.614
average total	0.581/0.588	0.600/0.602

Table 28: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the length and the time of the tweets.

Filtering By Amount, Length and Time

	Last 5 Years, Maximum Tweet Length and ... Amount of Tweets				
Corpus - Test Set	<100	100-1000	>1000	>100SS	>1000SS
M - MEN	0.609/0.612/31.7	0.700/0.695/17.2	0.682/0.700/14.9	0.707/0.716/14.1	0.713/0.712/11.6
M - SimLex999	0.201/0.198/6.5	0.256/0.250/0.5	0.277/0.268/0.2	0.291/0.284/0.3	0.314/0.301/0.2
M - 353 Sim	0.593/0.590/11.8	0.679/0.697/1.0	0.728/0.729/1.0	0.683/0.682/0.0	0.731/0.732/0.0
M - 353 Related	0.533/0.546/10.3	0.608/0.612/1.2	0.628/0.633/2.0	0.614/0.626/0.8	0.626/0.636/0.4
average	0.484/0.487	0.561/0.564	0.579/0.583	0.574/0.577	0.596/0.595
T - MEN	0.608/0.616/30.9	0.701/0.707/15.6	0.710/0.718/13.5	0.723/0.729/12.7	0.731/0.736/11.6
T - SimLex999	0.182/0.181/5.8	0.266/0.258/0.4	0.288/0.274/0.2	0.304/0.292/0.1	0.313/0.300/0.1
T - 353 Sim	0.584/0.566/14.3	0.714/0.720/2.0	0.693/0.692/1.0	0.674/0.670/0.5	0.706/0.712/0.0
T - 353 Related	0.517/0.501/11.5	0.608/0.620/2.4	0.632/0.634/1.6	0.605/0.610/1.2	0.635/0.649/0.4
average	0.473/0.466	0.573/0.576	0.581/0.580	0.576/0.575	0.596/0.599
V - MEN	0.600/0.599/30.4	0.689/0.696/14.0	0.715/0.721/12.7	0.714/0.718/11.8	0.722/0.726/9.8
V - SimLex999	0.166/0.168/5.5	0.279/0.267/0.4	0.291/0.282/0.0	0.305/0.290/0.0	0.309/0.293/0.0
V - 353 Sim	0.624/0.628/12.8	0.717/0.720/3.0	0.721/0.721/1.0	0.732/0.737/1.0	0.751/0.753/1.0
V - 353 Related	0.493/0.504/10.7	0.648/0.656/3.6	0.631/0.656/1.2	0.660/0.671/1.6	0.668/0.680/1.2
average	0.471/0.475	0.583/0.585	0.590/0.595	0.603/0.604	0.613/0.613
average total	0.476/0.476	0.572/0.575	0.583/0.586	0.584/0.585	0.602/0.602

Table 29: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the length, the amount, and the time of the tweets. (1)

	Last 5 Years, Minimum Tweet Length and ... Amount of Tweets				
Corpus - Test Set	<100	100-1000	>1000	>100SS	>1000SS
M - MEN	0.547/0.549/26.9	0.688/0.700/13.9	0.662/0.672/12.1	0.670/0.681/14.0	0.699/0.711/10.4
M - SimLex999	0.168/0.167/6.0	0.238/0.230/0.2	0.247/0.248/0.0	0.238/0.232/0.2	0.265/0.260/0.0
M - 353 Sim	0.558/0.555/13.8	0.707/0.719/0.0	0.678/0.686/1.5	0.709/0.727/0.5	0.709/0.715/0.0
M - 353 Related	0.501/0.503/11.9	0.629/0.634/1.2	0.607/0.609/2.4	0.621/0.626/1.2	0.631/0.649/1.2
average	0.444/0.444	0.566/0.571	0.545/0.554	0.556/0.567	0.576/0.584
T - MEN	0.513/0.518/27.5	0.679/0.695/12.2	0.669/0.678/10.9	0.687/0.700/13.0	0.697/0.706/9.5
T - SimLex999	0.191/0.183/7.9	0.224/0.223/0.1	0.254/0.250/0.0	0.239/0.233/0.1	0.272/0.266/0.0
T - 353 Sim	0.519/0.523/16.3	0.702/0.708/1.5	0.693/0.704/0.0	0.660/0.663/1.0	0.720/0.727/0.0
T - 353 Related	0.432/0.445/15.9	0.637/0.658/3.6	0.625/0.632/1.2	0.648/0.656/2.8	0.642/0.649/0.4
average	0.414/0.417	0.561/0.571	0.560/0.566	0.559/0.563	0.583/0.587
V - MEN	0.533/0.533/27.1	0.691/0.702/12.7	0.693/0.702/11.7	0.697/0.707/12.1	0.720/0.730/8.7
V - SimLex999	0.138/0.143/8.8	0.227/0.221/0.3	0.269/0.265/0.0	0.223/0.220/0.3	0.283/0.275/0.0
V - 353 Sim	0.594/0.581/18.7	0.699/0.694/1.5	0.694/0.700/1.5	0.684/0.693/1.5	0.696/0.696/1.0
V - 353 Related	0.457/0.446/13.9	0.612/0.620/2.8	0.658/0.687/2.4	0.638/0.647/2.4	0.628/0.650/0.8
average	0.431/0.426	0.557/0.559	0.579/0.589	0.561/0.567	0.582/0.588
average total	0.429/0.429	0.561/0.567	0.562/0.569	0.560/0.565	0.581/0.586

Table 30: Pearson correlation coefficient r / Spearman correlation coefficient ρ / percentage of missing words for each test set filtering by the length, the amount, and the time of the tweets. (2)

7.2 Complete Tables for Semantic Change

Best Split For All Basic Filters

Model - Filter	Split	MT	MV	TV	Sum	Avg
Full Corpus (Preprocessing but No Filters)						
Full	72-8	0.525	0.500	0.525	1.55	0.517
Raw Corpus (No Preprocessing and no Filters)						
Raw	63-17	0.538	0.538	0.538	1.612	0.538
Amount of Tweets per User						
<100 Tweets	9-67	0.526	0.553	0.526	1.605	0.535
$100 - 1000$ Tweets	35-45	0.562	0.538	0.462	1.562	0.521
>1000 Tweets	24-56	0.575	0.600	0.500	1.675	0.558
>100 SS Tweets	14-66	0.55	0.500	0.525	1.575	0.525
>1000 SS Tweets	19-61	0.588	0.562	0.538	1.688	0.562
Length						
25	29-51	0.612	0.562	0.538	1.712	0.571
50	23-57	0.612	0.562	0.512	1.688	0.562
75	43-37	0.512	0.562	0.538	1.612	0.538
Mean Length	29-51	0.588	0.588	0.488	1.662	0.554
Max Length	3-77	0.512	0.512	0.512	1.537	0.512
Min Length	50-30	0.500	0.575	0.55	1.625	0.542
Time						
First 5 Years	10-64	0.554	0.500	0.527	1.581	0.527
Last 5 Years	29-51	0.538	0.562	0.538	1.638	0.546

Table 31: Accuracy score, sum, and average for all three comparisons of the best possible split for each filter.

Best Split For All Combinations of Filters

Model - Filter	Split	MT	MV	TV	Sum	Avg
Combination - Length and Amount						
Max Length × <100 Tweets	34-20	0.593	0.519	0.556	1.667	0.556
Max Length ×100 − 1000 Tweets	34-46	0.550	0.525	0.475	1.550	0.517
Max Length × >1000 Tweets	0-79	0.506	0.506	0.506	1.519	0.506
Max Length × >100 SS Tweets	9-71	0.512	0.538	0.512	1.562	0.521
Max Length × >1000 SS Tweets	3-77	0.512	0.512	0.512	1.537	0.512
Min Length × <100 Tweets	38-9	0.681	0.638	0.638	1.957	0.652
Min Length ×100 − 1000 Tweets	8-72	0.550	0.525	0.525	1.600	0.533
Min Length × >1000 Tweets	18-62	0.550	0.500	0.500	1.550	0.517
Min Length × >100 SS Tweets	30-50	0.475	0.550	0.550	1.575	0.525
Min Length × >1000 SS Tweets	60-20	0.525	0.550	0.500	1.575	0.525
Combination - Length and Time						
Max Length × Last 5 Years	2-78	0.500	0.5 00	0.525	1.525	0.508
Min Length × Last 5 Years	42-38	0.525	0.525	0.500	1.550	0.517
Combination - Time and Amount						
Last 5 Years × <100	35-40	0.560	0.560	0.507	1.627	0.542
Last 5 Years × 100-1000	28-52	0.525	0.500	0.525	1.550	0.517
Last 5 Years × >1000	33-47	0.512	0.588	0.512	1.612	0.538
Last 5 Years × >100 SS	2-78	0.500	0.500	0.525	1.525	0.508
Last 5 Years × >1000 SS	27-53	0.562	0.588	0.538	1.688	0.562
Combination - Length and Time and Amount						
Max Length × Last 5 Years × <100 Tweets	0-54	0.519	0.519	0.519	1.556	0.519
Max Length × Last 5 Years × 100-1000 Tweets	1-79	0.512	0.512	0.512	1.537	0.512
Max Length × Last 5 Years × >1000 Tweets	0-79	0.506	0.506	0.506	1.519	0.506
Max Length × Last 5 Years × >100 SS Tweets	1-79	0.512	0.512	0.512	1.537	0.512
Max Length × Last 5 Years × >1000 SS Tweets	1-79	0.512	0.488	0.512	1.512	0.504
Min Length × Last 5 Years × <100 Tweets	24-15	0.564	0.667	0.564	1.795	0.598
Min Length × Last 5 Years × 100-1000 Tweets	17-63	0.538	0.562	0.512	1.612	0.538
Min Length × Last 5 Years × >1000 Tweets	59-21	0.488	0.538	0.488	1.512	0.504
Min Length × Last 5 Years × >100 SS Tweets	1-79	0.512	0.512	0.488	1.512	0.504
Min Length × Last 5 Years × >1000 SS Tweets	35-45	0.562	0.512	0.512	1.588	0.529

Table 32: Accuracy score, sum, and average for all three comparisons of the best possible split for each combination of filters.