Institut für Maschinelle Sprachverarbeitung

Universität Stuttgart
Pfaffenwaldring 5b
70569 Stuttgart

Bachelorarbeit

Lexical Semantic Change Discovery

Sinan Kurtyigit

Studiengang:	B.Sc. Informatik
Prüfer:	Apl. Prof. Dr. Sabine Schulte im Walde
Betreuer:	Dominik Schlechtweg
begonnen am:	01.09.2020
beendet am:	01.03.2021

Contents

Abstract

The field of Lexical Semantic Change Detection (LSCD) deals with the detection of words that change their meaning over time. While there is a large amount of research in the field, only few go beyond a standard benchmark evaluation of existing models. The goal of this thesis is to derive a practical benefit from previous milestones in research. Therefore, a framework is built, that utilizes common approaches for LSCD to discover novel changing words. The framework is highly automated and easily applicable, making it useful for both beginners and experienced users. Anyone, who has access to two corpora (e.g., from different time periods) can use this framework to automatically discover words that change their meaning between the two corpora. In an exemplary discovery process, which includes multiple fine-tuning phases on common tasks, the framework and its underlying discovery process are demonstrated. The framework is successfully used to discover changing words between two time-specific German corpora. Additionally, in the fine-tuning phases the framework is also used to evaluate and optimize the implemented approaches and model parameters. The results show that the framework provided in this thesis and its implemented approaches can be used for the discovery of novel changing words and also evaluation.

Das Gebiet LSCD (Lexical Semantic Change Detection) beschäftigt sich mit der Erkennung von Wörtern, die ihre Bedeutung im Laufe der Zeit ändern. Es gibt zwar eine große Anzahl an Forschungsarbeiten auf diesem Gebiet, aber nur wenige gehen über eine Standard-Benchmark-Evaluation bestehender Modelle hinaus. Das Ziel dieser Arbeit ist es, aus den bisherigen Meilensteinen der Forschung einen praktischen Nutzen abzuleiten. Dazu wird ein Framework erstellt, das gängige Ansätze für LSCD nutzt, um Wörter zu entdecken die einen Bedeutungswandel durchmachen. Das Framework ist hochgradig automatisiert und leicht anwendbar, wodurch es sowohl für Anfänger als auch für erfahrene Benutzer nützlich ist. Jeder, der Zugriff auf zwei Korpora (z.B. aus unterschiedlichen Zeiträumen) hat, kann dieses Framework nutzen, um automatisch Wörter zu entdecken, die ihre Bedeutung zwischen den beiden Korpora ändern. In einem exemplarischen Entdeckungsprozess, der mehrere

Feinabstimmungsphasen beinhaltet, wird das Framework und der zugrundeliegende Entdeckungsprozess demonstriert. Das Framework wird erfolgreich genutzt um Wörter zu entdecken die einen Bedeutungswandel zwischen zwei deutschen zeitspezifischen Korpora durchmachen. Zusätzlich wird das Framework in den Feinabstimmungssphasen auch zur Evaluierung und Optimierung der implementierten Ansätze und Modellparameter eingesetzt. Die Ergebnisse zeigen, dass das in dieser Arbeit vorgestellte Framework und seine implementierten Ansätze sowohl für die Entdeckung von Wörtern die einen Bedeutungswandel durchmachen als auch für die Evaluation verwendet werden können.

1 Introduction

Words change their meaning over time. While some senses are lost, other novel senses are established (Blank, 1997; p. 113). One such example is the German word *Aufkommen*. Previously, the word was used as ‘emergence’ as in ‘It is true that with the **emergence** of the manufactory, traces of child exploitation are showing.‘ Later, this sense was lost, and a new one was gained. The word was used as ‘production’ as in ‘They know that we need more feed from our own **production** for the cattle’. The automatic detection (with the help of computers), of such semantic changes has gained increasing importance over the years and consequently, the field of Lexical Semantic Change Detection (LSCD) has emerged (Kutuzov et al., 2018; Tahmasebi et al., 2018; Hengchen et al., 2021). Common tasks include creating data sets or developing, evaluating and analyzing models that detect change. In recent years, considerable progress has been made. Through extensive research the field now owns standard corpora and tuning data for different languages as well as optimized models for LSCD. However, only a limited amount of work applies the methods to discover novel instances of semantic change, rather than detecting semantic changes on a small set of words (Schlechtweg et al., 2020; Basile et al., 2020). Thus, the practical applicability has not been exploited yet. The goal of this thesis is to **derive a practical benefit from previous milestones in research and make lexical semantic change useful**.

For this, a framework that utilizes common approaches for LSCD to automatically discover novel changes is build. The discovery process is fully automated and easily applicable. Anyone, who has access to two corpora (e.g., from different time periods, domains or genres) and wants to discover words that undergo a change of meaning between them, can utilize this framework without much effort. Additionally, different tools are provided for fine-tuning and analyzing the implemented approaches. The framework can act as an entry point for newcomers to the field, by allowing them to easily generate their first results and further experiment with them to get comfortable with common approaches. Experienced users can use the fine-tuning and analyzing tools to aid their research. Furthermore, the full automation and ease of use could also assist people outside of the field, e.g., lexicographers, in their work, without the need of an extensive knowledge in LSCD and programming.

The thesis is split into two main parts. The first part (Section 3) describes the framework, its implemented methods and the underlying discovery process. Afterwards (in Section 4), the possible applications and the quality of the framework are illustrated by an exemplary discovery process using the German SemEval-2020 (Schlechtweg et al., 2020) data set. This includes multiple fine-tuning phases on different tasks to present the different features of the framework and a human annotation phase to evaluate the quality of the discovery.

1.1 Paper Submission

A scientific paper based on the thesis results (Section 4) was submitted in collaboration with Maike Park, Dominik Schlechtweg, Jonas Kuhn and Sabine Schulte im Walde. I wrote most parts of the paper myself, while receiving feedback from Dominik and Sabine; the description of the annotation and the WUGs had however been contributed by Dominik and have been taken over as annotation section (see 4.3) into my thesis.

2 Related Work

State-of-the-art semantic change detection models are Vector Space Models (VSMs) (Schlechtweg et al., 2020). These can be divided into static (type-based) (Turney and Pantel, 2010; Levy et al., 2015; Mikolov et al., 2013b) and contextualized (token-based) (Schütze, 1998; Devlin et al., 2019; Peters et al., 2018) approaches. Static embedding models are often used to solve word similarity and analogy tasks (Levy et al., 2015) and as ingredients for models solving downstream tasks (e.g. Hätty et al., 2020; Kutuzov et al., 2017). Static embedding models are the state-of-the-art models for LSCD across time and potentially across any type of domains (Hamilton et al., 2016; Schlechtweg et al., 2019; Hätty et al., 2019). Prominent static models include low-dimensional embeddings such as Global Vectors (GloVe, Pennington et al., 2014) and Skip-Gram with Negative Sampling (SGNS, Mikolov et al., 2013a;b). However, as these models come with the deficiency that they aggregate all senses of a word into a single representation, contextualized embeddings have been proposed (Peters et al., 2018; Devlin et al., 2019). According to Hu et al. (2019) these models can ideally capture complex characteristics of word use, and how they vary across linguistic contexts.

While large amounts of research exists, previous work mostly focuses on creating data sets or developing, evaluating and analyzing models. Contrary to this, the goal of the framework developed in this thesis, is to find 'undiscovered' changing words. Few studies (Kim et al., 2014; Hamilton et al., 2016; Takamura et al., 2017) focus on this task. Common and well-performing LSCD approaches are implemented in a framework to make a practical use of previous milestones in research. The discovery process is fully automated and easy to execute in order to allow users with different levels of experience to make large-scale discoveries on all kinds of corpora. Additionally, tools for evaluating, analyzing and further optimizing the implemented approaches can be utilized by researchers to reach new milestones.

3 Framework

The goal of the framework is to solve the task of **lexical semantic change discovery**:

> Given a corpus pair (C_1, C_2), decide for the intersection of their vocabularies which words lost or gained sense(s) between C_1 and C_2.

Discovery is an important task, with applications e.g. in lexicography where dictionary makers aim to cover the full vocabulary of a language. However, the task comes with difficulties. While the LSCD models are commonly used to detect changes for a small list of pre-selected target words (Schlechtweg et al., 2020; Basile et al., 2020), the intersection of two corpus vocabularies contains a much larger amount of words.

For some models (e.g., BERT), the computational effort rises drastically with the size of the vocabulary intersection. Furthermore, the intersection will likely contain many faulty words (e.g., misspelled words and foreign language), that stem from difficulties in the corpus creation (e.g., digitalization errors).

While the main focus of the framework is the automated discovery of changing words, scripts are provided to automatically solve the binary classification and graded ranking subtasks from SemEval-2020 Task 1 (Schlechtweg et al., 2020) (see Section 4.1). The framework also provides methods to evaluate the quality of the produced results, i.e., calculating evaluation metrics for the classification problem and calculating the Spearman rank-order correlation coefficient ρ.[1] This allows users to fine-tune the implemented models on their own data sets to find well-performing parameter configurations.

The core components of the framework are written in the Python programming language. Shell scripts are provided to fully automate the process. The framework is built modularly to facilitate debugging, analyzing and modifying the code, resulting in more clarity for both the developer and user. Furthermore, this allows the user to only use single parts of the framework when needed. The python scripts as well as the shell scripts contain a wide range of parameters, to incentivize experimenting with the provided approaches, and further optimizing these. Additionally, for the ease of use, a set of recommended parameters is provided.

3.1 Discovery Process

The following steps are executed to solve the task of lexical semantic change discovery:[2]

1. A neural language model (SGNS, BERT) is used to generate word embeddings (dense representation of words in the form of a numeric vector) for words in

[1] This requires gold data, e.g., human-annotated labels.

[2] Both approaches require additional model-specific steps, which are explained in the according sections.

the intersection of the corpus vocabularies.[3]

2. Differences between word embedding(s) from C_1 and word embedding(s) from C_2 are measured, resulting in graded values.

3. A threshold is calculated according to these graded values. Words whose graded values are greater than or equal to this threshold, are labeled as changing words.

4. A filtering is applied to these changing words in order to remove undesirable words (e.g., proper names and foreign words).

5. (Optional) The usages for the filtered changing words are extracted and stored in a specific format. These can then be used to evaluate the predictions or detect false positives.

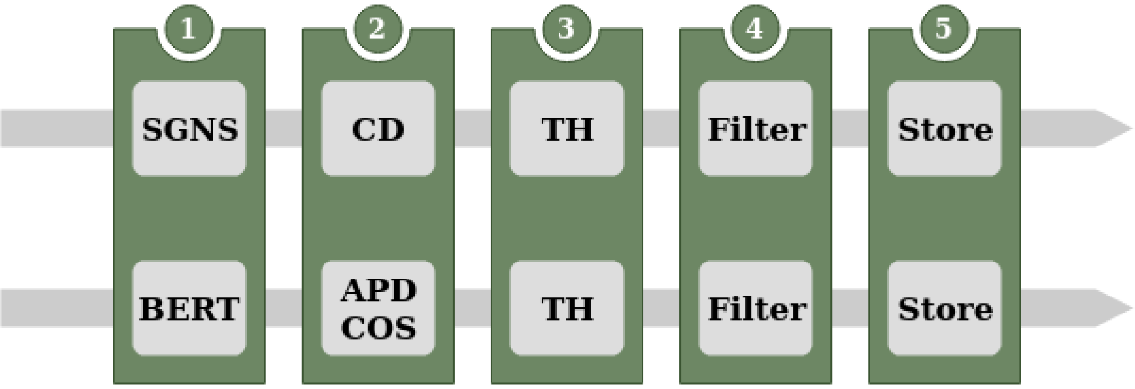


Figure 1: The essential steps of the discovery process.

3.2 Models

The framework provides a static and a contextualized model to generate word embeddings. While static models generate a single word embedding for a word, contextualized models generate a word embedding for every word usage, i.e., sentence where the word occurs.

[3] In BERT's case only a sample of the intersection is considered due to computational limitations.

3.2.1 Static Model

Most static approaches in LSCD combine three sub-systems (Schlechtweg et al., 2019) to generate graded values:

1. Creating word embeddings.
2. Aligning them across corpora.
3. Measuring differences between the aligned embeddings.

Creating Static Word Embeddings The implementation of the Skip-gram with Negative Sampling model (SGNS, Mikolov et al., 2013a;b) in Schlechtweg et al. (2019) is used to generate static word embeddings.[4]

SGNS is a shallow neural language model trained on pairs of word co-occurences extracted from a corpus. In short, given a vocabulary V, a target word x and a word from the vocabulary $v \in V$, the neural network learns how likely it is for the word v to appear in the context of the target word x. Figure 2 shows the structure of the skip-gram model. The network consists of an input layer, a single hidden layer (without an activation function) and an output layer. The input layer and hidden layer are connected by weights which are stored in a weight matrix W. Similarly, the weights connecting the hidden layer to the output layer are stored in a second weight matrix W'. The neural network is trained by feeding it word pairs found in the corpus. A word pair consists of the target word and another word appearing in its context. The context is defined by a symmetric windows size parameter (e.g., a window size of two considers two words to the left and two words to the right). For each training pair, all weights are adjusted slightly so that the network predicts the training sample more accurately. The input for the network is a one-hot vector with $|V|$ components. Every position is labeled as 0, except the one corresponding to the word, which is labeled as 1. The hidden layer performs the dot product between the weight matrix W and the input vector $\vec{x}$. Since the input vector is a one-hot

[4] The implementation is based on the gensim word2vec library (Řehůřek and Sojka, 2010).

vector, this simply selects the matrix row of the corresponding word. This vector is then passed to the output layer. Now the output layer computes the dot product between this output vector of the hidden layer and the weight matrix (W') of the output layer. Finally, the softmax activation function is applied to compute the output vector $\vec{y}$. The output vector is also a single vector with $|V|$ components. Every positions contains the probability that the corresponding word occurs in the context of the target word x. As mentioned, all weights are adjusted slightly for every training sample. In order to lessen the huge computational effort caused by this, negative sampling is used (Mikolov et al., 2013b). The idea is that every training sample only modifies a small percentage of the weights, rather than all of them. A small number (according to the negative sampling parameter k) of "negative" words (words for which we want the network to output a 0) are selected. Additionally the "positive" word is selected. Now only the weights for these $k+1$ words are updated. Once the model is fully trained, only the first weight matrix W is of interest. The optimized weight vectors can be interpreted as a semantic vector space that contains the embeddings for all words in the vocabulary.

Following standard practice, both spaces are length-normalized and mean-centered (Artetxe et al., 2016; Schlechtweg et al., 2019) to optimize the word embeddings. The implementation in Schlechtweg et al. (2019) is used for this.[5]

Alignment The resulting vector spaces containing the word embeddings for C_1 and C_2 respectively, are then aligned by applying Orthogonal Procrustes (OP), because columns from different vector spaces may not correspond to the same coordinate axes (Hamilton et al., 2016). For this, the implementation of Artetxe et al. (2018) is used.

Measuring Differences Given two word embeddings $\vec{w}_1$ and $\vec{w}_2$, the difference is measured by calculating their Cosine Distance (CD) (Salton and McGill, 1983):

$$CD(\vec{w}_1, \vec{w}_2) = 1 - cos(\vec{w}_1, \vec{w}_2), \tag{1}$$

[5] The implementation is based on SciPy (Virtanen et al., 2020) and NumPy (Harris et al., 2020).

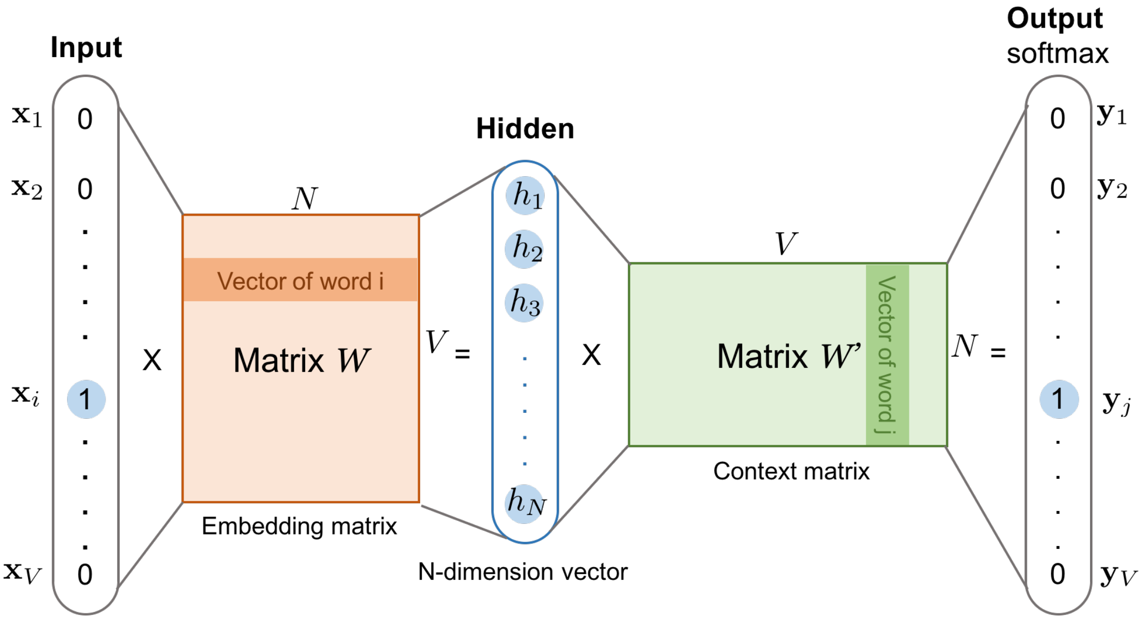


Figure 2: Structure of the skip-gram model (Weng, 2017).

where cos is the cosine of the angle between $\vec{w}_1$ and $\vec{w}_2$. The CD can take values between -1 (exactly rectified) to 1 (exactly opposed). However, the weights in the word vectors are non-negative and thus, in this use case the CD ranges from 0 (related) to 1 (unrelated). A value close to 1 indicates that word has undergone a semantic change and a value close to 0 indicates the opposite (no change). Again, the implementation in Schlechtweg et al. (2019) is used.[6]

The sub-system combination of SGNS+OP+CD has performed very well in recent shared tasks (Schlechtweg et al., 2020; Basile et al., 2020), ranking among the best submissions (Arefyev and Zhikov, 2020; Kaiser et al., 2020b; Pömsl and Lyapin, 2020; Pražák et al., 2020).

[6] The implementation is based on SciPy (Virtanen et al., 2020).

3.2.2 Contextualized Model

Contextualized models take word usages as input to generate contextualized embeddings. Therefore, the procedure of generating graded values is slightly different:

1. Sample words that will act as an input for the model.
2. Extract usages from the corpora for the sampled words.
3. Create two sets of contextualized word embeddings for every word in the sample.
4. Measure differences between the two sets of embeddings.

Sample from the Intersection of the Vocabularies Contextualized models generate a word embedding for every usage of a word. Unfortunately, both the extraction of usages as well as the generation of word embeddings with the contextualized model are computationally expensive. Depending on the size of the intersection of the corpus vocabularies, these tasks become infeasible on normal hardware. Therefore, words from this intersection have to be sampled. This sample then acts as a pool of possible changing words and is also needed to calculate the threshold. A sample size of 500 words is recommended as a good balance between feasibility and number of predictions. However, a variable size parameter allows end-users with more resources to increase the size of the sample at will.

Zipf's law (Zipf, 1935; 1949) states that given a corpus, the frequency (number of occurrences) of any word is inversely proportional to its rank in a frequency ranking. Thus, the second most frequent word will only occur half as often as the most frequent one, the third most frequent word will only occur a third as often, etc. This means that a overwhelming majority of the words in a corpus are low-frequent ones. Therefore, a random sampling is likely to not contain any higher-frequency words. In order to have predictions not only for low-frequency words, the following sampling procedure is implemented:

1. Compute the frequency range (highest frequency - lowest frequency) of the vocabulary intersection.
2. Split this range into five areas of equal frequency width.
3. Take random samples from these areas according to how many words they contain.

Consider the following example:

(2) Given a sample size of 500, a lowest frequency of 1 and a highest frequency of 100:

   1. The frequency range equals 99 ($= 100 - 1$).
   2. This range is then split into five areas of width 20 ($= \lceil 99/5 \rceil$). The first area contains the words with a frequency less than or equal to 20, the second area contains the words with a frequency higher than 20 and less than or equal to 40, etc.
   3. Let the first area contain 50%, the second area 30%, the third area 10%, the fourth area 8% and the fifth area 1% of the words from the intersection. Then 250, 150, 50, 40 and 10 words are randomly sampled from the respective areas.

Extracting usages A script is provided, that given a list of words (e.g., a sample of the vocabularies intersection) automatically extracts usages from the corpora. The user can choose the maximum number of usages to extract from a corpus. The usages are then randomly extracted from both corpora and stored in an appropriate format. For users that have access to two pairs of corpora, i.e., a raw and a lemmatized pair, the script will extract sentences for both corpora pairs and save them accordingly.[7]

[7] These can be used for the pre-processing approaches that are described in the next paragraph.

Creating Contextualized Word Embeddings The implementation of the Bidirectional Encoder Representations from Transformers (BERT, Devlin et al., 2019) model in Laicher et al. (2020) is used to generate contextualized word embeddings.[8]

BERT is a transformer-based (Vaswani et al., 2017) neural language model designed to find contextualized representations by analyzing left and right contexts. The training objective is to solve two unsupervised tasks simultaneously and minimize their combined loss function:

1. **Masked Language Model (MLM)**: Randomly replace 15% of the words in the input by a [MASK] token. Predict the masked words based on the context provided by the other non-masked words.
2. **Next Sentence Prediction (NSP)**: Given two sentences A and B, predict whether B is the actual sentence that comes after A, or just a random sentence from the corpus.

The idea behind the MLM task is that the model looks at both directions (using left and right contexts) to predict the masked word, making it deeply bidirectional. The second task is solved simultaneously, in order to better understand the relationship between two sentences.

The key part is the self-attention mechanism, which is used to find the words of importance for the words in the input sequence. First, the input, i.e., a sentence or two sentences, is split into tokens. These tokens are then mapped onto embeddings. Afterwards, these are passed to the self-attention head (see Figure 3). For every embedding, a key, query and value vector with 64 components is created by a matrix multiplication with the respective key, query and value matrices (Allamar). The following steps are executed for every token (Futrzynski):

1. The dot products between the token's query vector and the key vectors of all tokens are calculated.

[8] The implementation is based on SciPy (Virtanen et al., 2020) and transformers (Wolf et al., 2020).

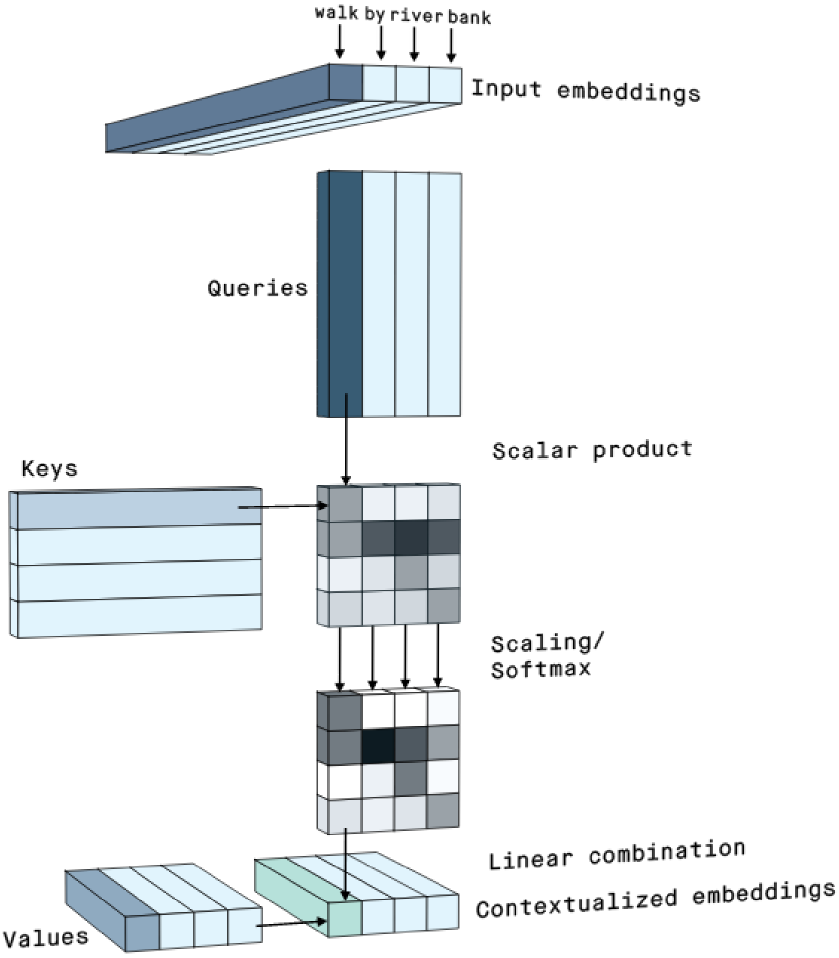


Figure 3: The Self-Attention Head (Futrzynski).

2. These values are normalized by a non-linear softmax activation function, resulting in attention scores.
3. A new (contextualized) embedding is created by linear combination of the value vector of all tokens, where the attention scores act as coefficients.

This results in transformed embeddings, which now encode information about their surroundings. This self-attention mechanism is run multiple times simultaneously (multi-head attention), with different key, query and value matrices. The embeddings from the different attention-heads are then concatenated together and passed to a feed-forward neural network. The *bert-base* model, that is used in this framework, has 12 self-attention-heads, resulting in contextualized embeddings with 768 components after concatenation. The model uses 12 layers of multi-head attention (see Figure 4), where the output from one layer is passed as an input to the next layer.

The contextualization is one of the key differences in comparison to the static SGNS model. Furthermore, while SGNS is trained from scratch on the task-specific data, BERT models, that are already pre-trained on large amounts of data, are already provided. The previously extracted usages are fed into BERT to encode the contextualized information onto the embeddings. These are then extracted from either one of the 12 different layers or as the average over multiple of those. A parameter allows the user to choose at will.

Following the success in Laicher et al. (2021), two pre-processing approaches are implemented:

- **Lemma**: Lemmatized usages are fed into BERT instead of raw usages.[9]
- **TokLem**: The target words in the raw usages are replaced by their lemma.

Measuring Differences Given two sets of word embeddings U_1 and U_2, two different approaches are implemented to measure differences:

- **Average Pairwise Distance (APD)**: First, if one set is larger, it is randomly downsampled so both sets contain the same number of word embeddings. Afterwards, for every possible pairing of word embeddings between both sets, the CD is calculated. The average over all CDs then corresponds to the change score (Schlechtweg et al., 2018; Giulianelli et al., 2020):

$$APD(U_1, U_2) = \frac{1}{|U_1| \cdot |U_2|} \sum_{u_1 \in U_1, u_2 \in U_2} CD(u_1, u_2). \tag{3}$$

[9] This is only possible if lemmatized corpora are provided.

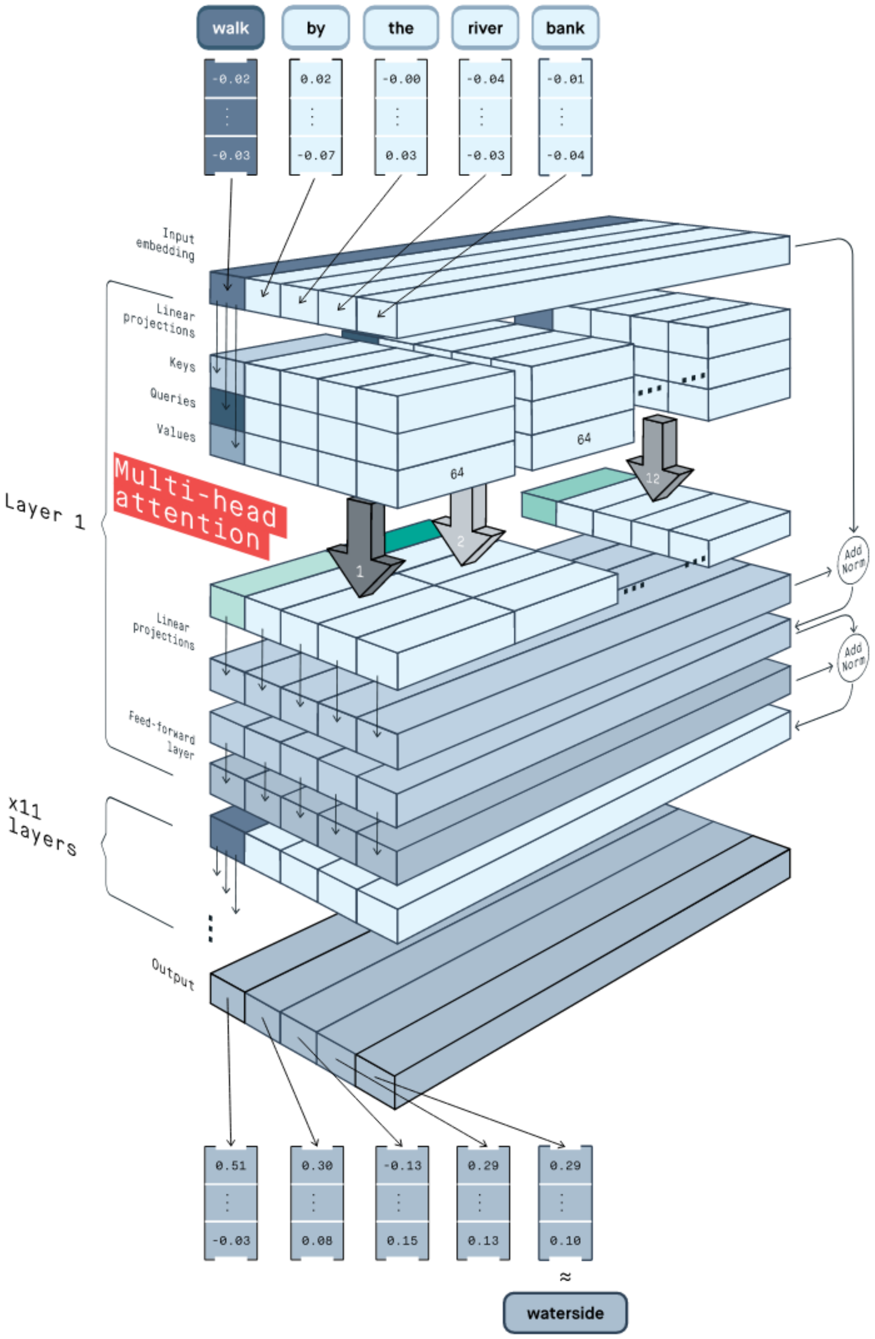


Figure 4: The Bert model (Futrzynski).

The APD values range from 0 (related) to 1 (unrelated). Values closer to 0 indicate that no semantic change happened between the sets of word embeddings, while values closer to 1 indicate a strong semantic change between the sets.

- **Cosine Similarity (COS)**: First, both sets are averaged respectively. Afterwards, the CD between the resulting mean embeddings is measured (Kutuzov and Giulianelli, 2020):

  (4) $$COS(U_1, U_2) = CD\left(\frac{\sum_{u_1 \in U_1} u_1}{|U_1|}, \frac{\sum_{u_2 \in U_2} u_2}{|U_2|}\right).$$

  Analogous to CD, COS can take values from 0 (related) to 1 (unrelated). Again, word whose meaning has not change between the sets, should result in smaller values, while words that have undergone a meaning change should result in larger values.

3.3 Thresholding

Thresholding is commonly used to infer binary values from graded ones (Schlechtweg et al., 2020; Basile et al., 2020). The idea is to define a threshold and rank every word based on its graded value. Words whose graded values are greater than or equal to this threshold, are marked as changing words, while words with a lower graded value are marked as non-changing words. When training data is available, i.e., a set of target words with human-labeled binary change scores, different thresholds can be evaluated to find a well-performing one. However, without training data, this becomes much more difficult. For such cases, Kaiser et al. (2020b) propose to choose the threshold according to the CDs of all words in the intersection of the corpus vocabularies. Motivated by this work, but with the addition of a variable parameter t for fine-tuning purposes, the following threshold is implemented:

(5) $$TH = \mu + t \cdot \sigma,$$

where μ is the mean and σ is the standard deviation of all graded values. Kaiser et al. show that the special case of $t = 1$ works very well with the SGNS model on the Italian data set provided by the DIACR-Ita shared task (Basile et al., 2020).

3.4 Filtering

The predicted changing words will likely contain proper names, foreign language and lemmatization errors. Since these are usually not considered as semantic changes, two filters are implemented with the help of spaCy (Honnibal et al., 2020), to remove such cases:

1. The first filter acts on a lemma-level and only allows nouns, verbs and adjectives to pass.[10]
2. The second filter operates on a usage-level. Words where over 10% of the usages do not correspond to the language or contain more than 25% punctuation are filtered out. Note, that the static approach requires usages for the words that passed the first filter. Depending on the amount of words the extraction process can be time-consuming. Therefore, for the static approach a parameter is provided, so users with low resources can apply the second filter to a sample of the words instead.

3.5 Store Usages for Human-Annotation

An optional step is provided to store the usages of the filtered changing words in a specific format. Falsely discovered changing words can then be removed by manually (human-based) inspecting the corresponding usages. This process can be assisted by the openly available DURel interface for annotation and visualization (see Section 4.3). The extracted usages are saved in a way so that they can be directly uploaded into the DURel interface. With the help of human-annotators, false positives can be found and removed. Additionally, the usages and the DURel interface can be used to evaluate the quality of the discovered changing words.

However, it should be noted that the manual human-annotation process is costly and can obviously not be automated. Therefore, it is not an essential part of the

[10] For BERT the first filter is applied to the intersection of the vocabularies before the random sample is taken, in order to not waste computational power on undesirable words.

discovery process, but rather an optional step for users with more resources, that want to further improve the quality of the discovered words.

4 Framework Application

In this section, a full discovery process is shown to illustrate the framework and it's possible applications. This process includes a fine-tuning of the model parameters and the threshold on the German SemEval-2020 data set by solving the two corresponding subtasks (Schlechtweg et al., 2020). All of this is done with tools provided by the framework.

Using the best-performing parameter configurations for both approaches, two sets of discovered changing words are generated. Afterwards, both sets are uploaded into a human-annotation system to evaluate their quality.

4.1 Data and Subtasks

The German data set provided by the SemEval-2020 shared task (Schlechtweg et al., 2020) is used for the discovery process. The data set contains a diachronic corpus pair for two time periods to be compared, a set of carefully selected target words as well as binary and graded gold data for semantic change evaluation and fine-tuning purposes.

Corpora The DTA corpus (Deutsches Textarchiv, 2017) and a combination of the BZ (Berliner Zeitung, 2018) and ND (Neues Deutschland, 2018) corpora are used. DTA contains texts from different genres spanning the 16th–20th centuries. BZ and ND are newspaper corpora jointly spanning 1945–1993. Schlechtweg et al. (2020) extract two time specific corpora C_1 (DTA, 1800–1899) and C_2 (BZ+ND 1946–1990) and provide a raw and a lemmatized version.

Target Words A list of 48 target words, consisting of 32 nouns, 14 verbs and 2 adjectives is provided. These are controlled for word frequency to minimize model biases that may lead to artificially high performance (Dubossarsky et al., 2017; Schlechtweg and Schulte im Walde, 2020).

Subtask 1: Binary Classification For a set of target words, decide which words lost or gained sense(s) between C_1 and C_2.

Subtask 2: Graded Ranking Rank a set of target words according to their degree of LSC between C_1 and C_2.

4.2 Tuning

The discovery process is closely related to the SemEval-2020 subtasks. And in fact, the task of discovering changing words can be seen as a special case of Subtask 1, where the list of target words equals the intersection of the corpus vocabularies. Hence, parameter configurations that perform well on these subtasks should result in good predictions. Therefore, both approaches are fine-tuned on the SemEval-2020 data.

First, Subtask 2 is solved to optimize the graded value predictions. Both models described above are used with multiple parameter configurations to generate sets of graded value predictions for the 48 SemEval target words. These are then evaluated by computing the Spearman rank-order correlation coefficient ρ between the graded value predictions and the graded gold data provided by the SemEval-2020 data set. The Spearman correlation coefficient ρ is a measure for the strength of association between two variables, whose values range from -1 (strong negative correlation) to 1 (strong positive correlation). A value close to 0 indicates no correlation between the variables. A strong positive correlation indicates a high performance.

Afterwards, the thresholding with t values ranging from -2 to 2 in steps of .1 is applied to the best-performing sets of graded values for both models. The

4: Identical
↑ 3: Closely Related
2: Distantly Related
1: Unrelated

Table 1: DURel relatedness scale (Schlechtweg et al., 2018)

binary gold data provided by the SemEval-2020 data set is then used to calculate precision, recall and $F_{0.5}$. Precision measures how many of the words that are labeled as changing words are indeed changing words. Recall measures how many of the existing changing words were labeled as such. The $F_{0.5}$-score considers both, but with a higher importance on precision. All three measures can take values from 0 to 1. A higher value indicates a better performance.

For both approaches, the parameter configuration with the highest peak $F_{0.5}$-score and the corresponding threshold is then chosen to discover changing words in a sample of 500 words.

4.3 Annotation

The model predictions are validated by human annotation. For this, the SemEval-2020 Task 1 procedure, as described in Schlechtweg et al. (2020), is applied. Annotators are asked to judge the semantic relatedness of pairs of word usages, such as the two usages of *Aufkommen* in (6) and (7), on the scale in Table 1.

(6) Es ist richtig, dass mit dem **Aufkommen** der Manufaktur im Unterschied zum Handwerk sich Spuren der Kinderexploitation zeigen.
'*It is true that with the* ***emergence*** *of the manufactory, in contrast to the handicraft, traces of child exploitation are showing.*'

(7) Sie wissen, daß wir für das Vieh mehr Futter aus eigenem **Aufkommen** brauchen.

‘*They know that we need more feed from our own* ***production*** *for the cattle.*’

The annotated data of a word is then represented in a Word Usage Graph (WUG, Schlechtweg et al., submitted), where vertices represent word usages and weights on edges represent the (median) semantic relatedness judgment of a pair of usages. The final WUGs are clustered with a variation of correlation clustering (Bansal et al., 2004; Schlechtweg et al., 2020) (see Figure 5, left) and split into two subgraphs representing nodes from C_1 and C_2 respectively (middle and right). Clusters are then interpreted as word senses and changes in clusters over time as lexical semantic change.[11]

In contrast to Schlechtweg et al. the openly available DURel interface for annotation and visualization is used.[12] This also implies a change in sampling procedure, as the system currently implements only random sampling of use pairs (without SemEval-style optimization). For each target word, 25 usages (sentences) per subcorpus (C_1, C_2) are sampled and uploaded to the DURel system, which presents use pairs to annotators in randomised order. Four German native speakers with university level education are recruited as annotators. Three have a background in linguistics, and one has an additional professional background in lexicography. Similar to Schlechtweg et al., the robustness of the obtained clusterings is ensured by continuing the annotation of a target word until all multi-clusters (clusters with more than one usage) in its WUG are connected by at least one judgment. Finally, a target word is labeled as changing (binary) if it gained or lost a cluster over time. For instance, *Aufkommen* in Figure 5 is labeled as change as it gains the orange cluster from C_1 to C_2.[13] Find an overview over the final set of WUGs in Table 3. A comparably high inter-annotator agreement (.67 Krippendorf’s α) is reached.

[11] The data set is available at `https://www.ims.uni-stuttgart.de/data/wugs`

[12] `https://www.ims.uni-stuttgart.de/data/durel-tool`.

[13] Following Schlechtweg et al. k and n are used as lower frequency thresholds to avoid that small random fluctuations in sense frequencies caused by sampling variability or annotation error be misclassified as change. $k = 1$ and $n = 3$ are set.

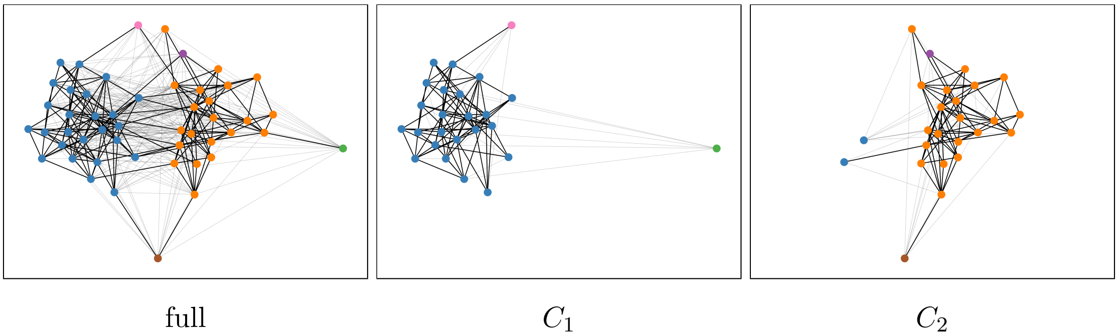


Figure 5: Word Usage Graph of German *Aufkommen* (left), subgraphs for first time period C_1 (middle) and for second time period C_2 (right). **black**/gray lines indicate **high**/low edge weights.

4.4 Results

In this section, the results of the tuning and discovery process are described.

4.4.1 Tuning

The SGNS model is commonly used in the field of LSCD (Schlechtweg et al., 2020) and already highly optimized (Kaiser et al., 2020a;b; 2021). Therefore, well-performing parameter configurations are known and difficult to further improve upon. Various parameter configurations based on the work in Kaiser et al. (2020a) are tested on the German SemEval-2020 data set.[14] The three best-performing configurations are presented in Table 2. These yield competitive $\rho = .690$, .710 and .710 respectively.

The performance of token-based approaches like BERT is drastically below the type-based counterparts in the SemEval-2020 shared task (Schlechtweg et al., 2020). However, Kutuzov and Giulianelli (2020) were able increase the performance immensely by fine-tuning these token-based models. Recently, Laicher et al. (2020;

[14] All configurations use $w = 10$, $d = 300$, $e = 5$ and a minimum frequency count of 39.

	parameters	t	**tuning**				**predictions**			
			ρ	$\mathrm{F}_{0.5}$	P	R	ρ	$\mathrm{F}_{0.5}$	P	R
SGNS	$k = 1$, $s = .005$	1.0	.690	.692	.750	.529				
	$\mathbf{k = 5, s = .001}$	1.0	.710	**.738**	.818	.529	.324	**.748**	.704	1.0
	$k = 5$, $s =$ None	1.0	.710	.685	.714	.588				
BERT	APD	-0.2	.673	.598	.560	.824				
	COS	1.0	.738	**.741**	.706	.788	.482	.620	.567	1.0

Table 2: Performance (Spearman ρ, $\mathrm{F}_{0.5}$-measure, precision P and recall R) of different approaches on tuning data (SemEval targets) as well as performance of best type- and token-based approach on respective predictions with optimal tuning threshold t.

2021) achieved competitive results on the SemEval-2020 data by experimenting with different layers and various pre-processing approaches. Following their work, the framework is used to test the performance of different layers and pre-processings. And indeed, by using the TokLem pre-processing in combination with layers 1+12, both APD and COS perform on a very high-level for Subtask 2 ($\rho = .690$ and .738).

The three best performing SGNS configurations, as well as the two BERT configurations (see Table 2) are considered for the second fine-tuning phase. After applying thresholding as described in Section 3, $F_{0.5}$-scores for a large range of thresholds are obtained. Table 2 presents the resulting peak $F_{0.5}$-score as well as the corresponding precision and recall for every configuration. SGNS achieves peak $F_{0.5}$-scores of .692, .738 and .685 respectively. Interestingly, the optimal threshold is at $t = 1.0$ in all three cases. This corresponds to the threshold used in Kaiser et al. (2020b). While the peak $F_{0.5}$ of BERT+APD is marginally worse (.598 at $t = -0.2$), BERT+COS is able to outperform the best SGNS configuration with a peak of .741 at $t = 0.1$.

4.4.2 Discovery

For both approaches, the top-performing configuration (see Table 2) is used to discover two sets of changing words. The filtering as described in Section 3.4 is applied to both sets. A set of 27 and a set of 75 words labeled as changing remain. 30 targets from the second set of changing words are randomly sampled to obtain a feasible number for annotation. The first set is called SGNS targets and the second one BERT targets, with an overlap of 6 targets. Following the annotation process, binary and graded gold data is generated for both target sets, in order to validate the quality of the discovery.

The evaluation is presented in Table 2. $\mathrm{F}_{0.5}$-scores of .748 for SGNS and .620 for BERT are achieved. Out of the 27 words predicted by the SGNS model, 19 (70 %) were actually labeled as changing words by the human annotators. In comparison, only 17 out of the 30 (56 %) BERT predictions were annotated as such. The performance of SGNS on the predictions (SGNS targets) is even higher than on the tuning data (SemEval targets). In contrast, BERT's performance on the predictions drops strongly in comparison to the performance on the tuning data (.741 vs. .620). This reproduces previous results and confirms that BERT generalises poorly for LSCD and does not transfer well between data sets (Laicher et al., 2020).

Figure 6 shows the detailed $F_{0.5}$ developments across different thresholds on the SemEval targets and the predicted words. Increasing the threshold on the predicted words improves the $F_{0.5}$ for both the type-based and token-based approach. A new high-score of .783 at $t = 1.3$ is achievable for SGNS. While BERT's performance also increases to a peak of .714 at $t = 1.0$, it is still lower than in the tuning phase.

4.4.3 Analysis

To find out what went wrong, false positives as well as their WUGs and underlying usages are inspected. Most of the wrong predictions can be grouped into one out of two error sources.

1. **Context Change**: The first category includes words where the context in

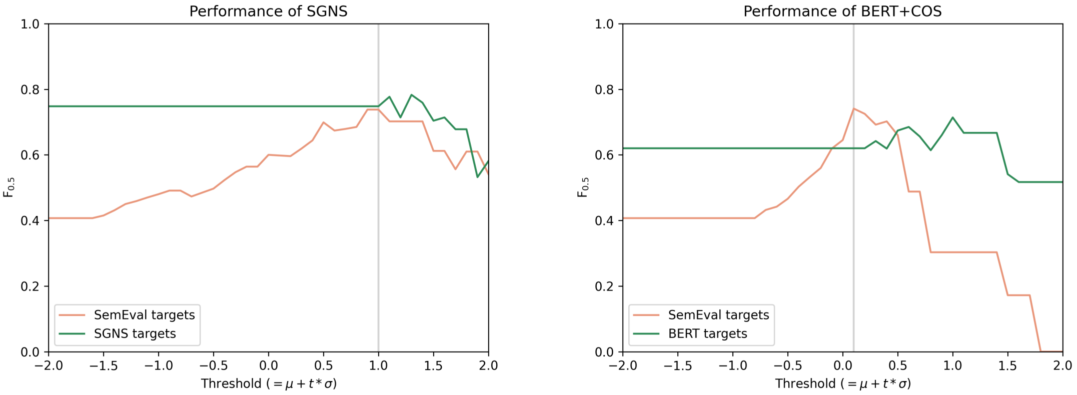


Figure 6: $F_{0.5}$ performance on SemEval targets (orange) and respective predictions (green) across different thresholds. Left: SGNS. Right: COS. Gray vertical line indicates optimal performance on SemEval targets.

the usages shifts between time periods, while the meaning stays the same. The WUG of *Angriffswaffe* (‘offensive weapon’) (see Figure 7) shows a single cluster for both C_1 and C_2. In the first time period *Angriffswaffe* is used to refer to a hand weapon (such as ‘sword’ or ‘spear’). In the second period, however, the context changes to nuclear weaponry. We can see a clear contextual shift, while the meaning did not change. In this case both models are tricked by the change of context. Further false positives in this category are the SGNS targets *Ächtung* (‘ostracism’) and *aussterben* (‘to die out’) and the COS targets *Königreich* (‘kingdom’) and *Waffenruhe* (‘ceasefire’).

2. **Context Variety**: Words that can be used in a large variety of contexts form the second group of false positives. SGNS falsely predicts *neunjährig* as a changing word. As seen in the corresponding WUG (see Figure 8), there is only one and the same cluster in both time periods, and the meaning of the target does not change, even though a large variety of contexts exists in both C_1 and C_2. For example: ‘which bears oats at **nine years** fertilization’, ‘courageously, a **nine-year-old** Spaniard did something’ and ‘after nine years of work’. Both

	General							Subtask 1/2	
	n	N/V/A	SPR	KRI	UNC	LOSS	JUD	LSC	LSC
SemEval	48	32/14/2	.59	.54	-	.20	38k	.35	.31
Predictions	51	28/10/13	.74	.67	.16	.22	14k	.63	.51
SGNS	27	12/4/11	.75	.68	.19	.22	9k	.70	.58
BERT	30	21/6/3	.71	.64	.14	.22	8k	.57	.48

Table 3: Overview target words. n = no. of target words, N/V/A = no. of nouns/verbs/adjectives, SPR = mean pairwise Spearman. KRI = Krippendorff's alpha. UNC = mean of uncompared multi-cluster combinations. LOSS = mean of normalized clustering loss * 10, JUD = no. of judged usage pairs, LSC = mean binary/graded change score.

models are misguided by this large context variety. Examples include the SGNS targets *neunjährig* ('9-year-old') and *vorjährig* ('of the previous year') and the COS targets *Bemerken* ('notice') and *durchdenken* ('to think through').

Lastly, consider two of the many words that are correctly labeled as changing by the SGNS model, and their corresponding WUGs. The uncommon word *Zehner* (see Figure 9), is a prime example of the framework's capabilities and the underlying approaches. While many clusters exist for the word, two of those stand out: the blue cluster and the orange one. In the first time-period, *Zehner* was used in a numerical sense, often in combination with *Hunderter* ('hundreds') und *Tausender* ('thousands'), such as in (8).

(8) Man sieht also, daß die Striche nach den Tausenden, nach den Hunderten und nach den **Zehnern** gesetzt werden.
'*So you can see that the strokes are placed after the thousands, after the hundreds, and after the* ***tens****.*'

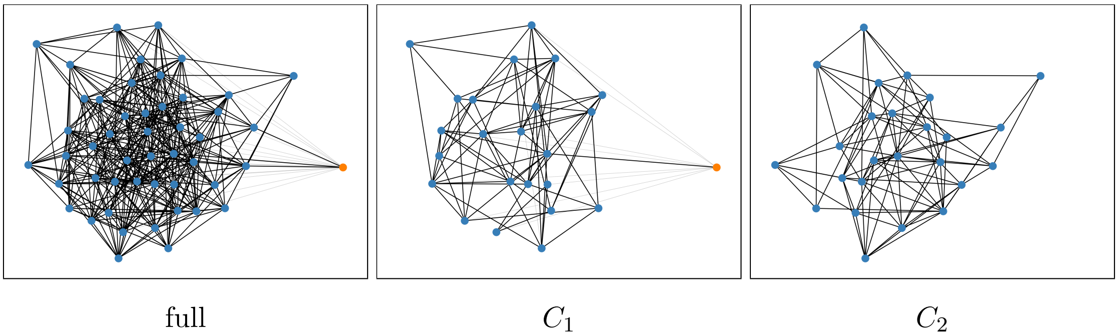


Figure 7: Word Usage Graph of German *Anriffswaffe* (left), subgraphs for first time period C_1 (middle) and for second time period C_2 (right).

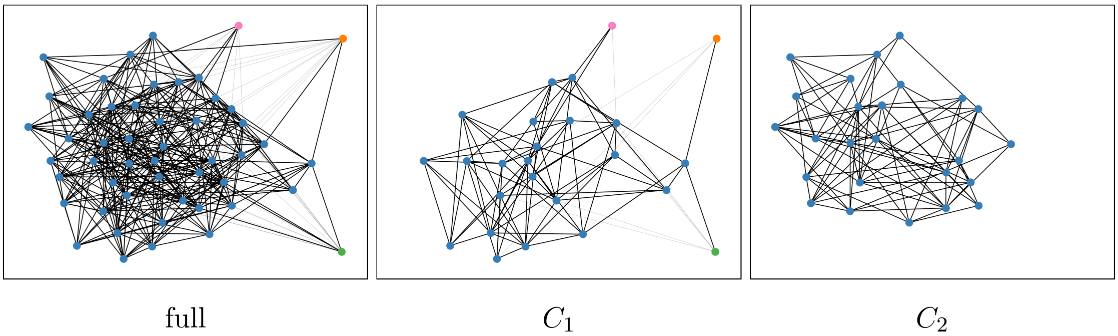


Figure 8: Word Usage Graph of German *neunjährig* (left), subgraphs for first time period C_1 (middle) and for second time period C_2 (right).

This meaning was lost over time and is nonexistent in the second time-period, as seen by the absence of the blue cluster in C_2. Interestingly, at the same time a novel word sense was gained. In C_2 a large orange cluster appeared, which was not present before. The usages show that, in the second time-period *Zehner* is used in the context of soccer lottery, as for example in (9).

(9) Fußball-Toto : Kein Elfer ; 6 **Zehner** mit je 3778 Mark ; 152 Neuner mit je 298 Mark.
'Soccer lottery : No eleven ; 6 ***tens*** *with 3778 marks each ; 152 nines with 298 marks each.'*

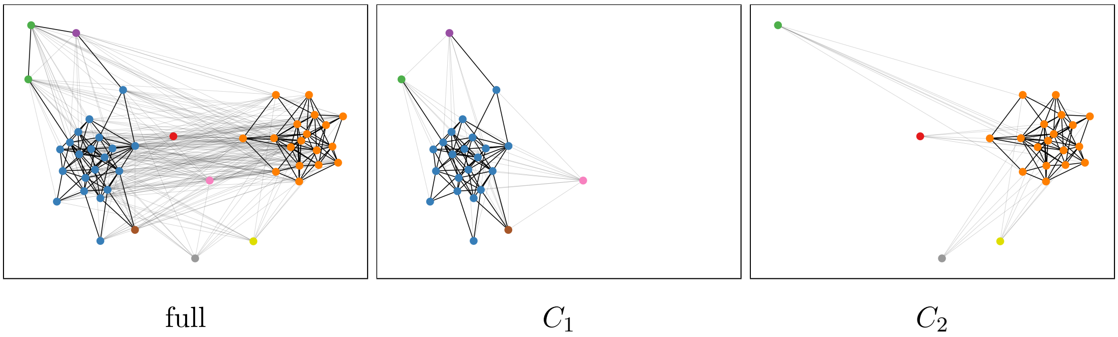


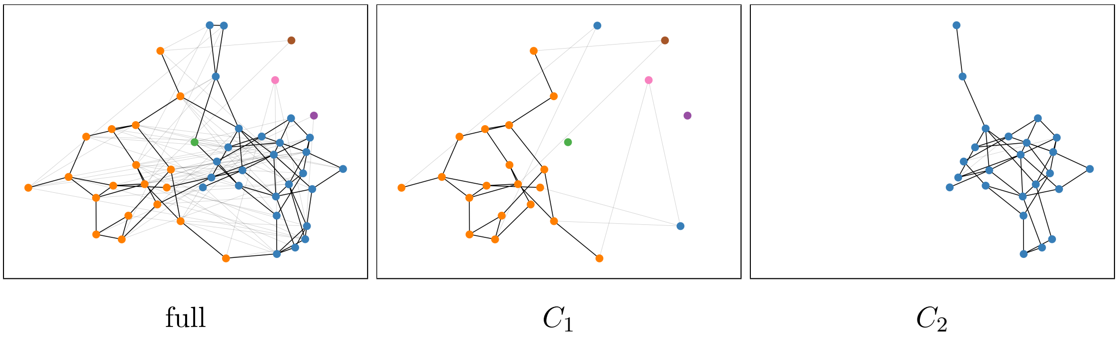
Figure 9: Word Usage Graph of German *Zehner* (left), subgraphs for first time period C_1 (middle) and for second time period C_2 (right).

The second word is *Sprachrohr*. Again, different clusters exist for the word, but the orange cluster and the blue cluster stand out. *Sprachrohr* was used to describe the physical device that directs the propagation of sound, improving intelligibility even at a further distance of the listener from the speaker. An exemplary usage is the following:

(10) Mittelst eines durch die Wand gehendes **Sprachrohrs**, wird der Heizer commandiert.
‘*Through a **voice tube** going through the wall, the stoker is commanded’*

This corresponds to the orange cluster. As seen in the WUG for C_2 the orange cluster disappeared, while the blue one emerged. In the second time-period *Sprachrohr* referred to the ‘spokesman’ of someone as in

(11) “Das Vaterland ist in Gefahr” - trommelte 1913 pausenlos der Wehrverein als **Sprachrohr** des Imperialismus.
‘*“The fatherland is in danger” - drummed the defense association as the **spokesman** of imperialism in 1913.’*



full C_1 C_2

Figure 10: Word Usage Graph of German *Sprachrohr* (left), subgraphs for first time period C_1 (middle) and for second time period C_2 (right).

5 Conclusion

The goal of thesis was to make LSCD useful by deriving practical applicability from previous research. Thus, a framework was build to automatically detect changing words. Additionally, different tools were implemented to solve tasks related to the field of LSCD. The framework should assist inexperienced users (e.g., beginners and people outside the field) with the large automation and ease-of-use as well experienced users, by providing generally applicable tools for analyzing and fine-tuning purposes.

Section 4 illustrated the complete discovery process, including multiple fine-tuning phases. Both approaches (static and contextualized) were used to successfully discover changing words, although the static model (SGNS) performed considerably better. The performance of SGNS is more stable for varying parameter configurations and SGNS generalizes better between data sets. Hence, SGNS is the recommended model to discover changing words. Furthermore, the results showed similarly to previous research that contextualized models like BERT can also perform well but a lot more fine-tuning is necessary. In the fine-tuning phase, the framework was also successfully used to solve tasks beyond LSC discovery. High performances were reached with both models for the SemEval-2020 Subtask 1 and Subtask 2.

The exemplary discovery process also showed some weaknesses of the framework. While both approaches find changing words, some are missed, and many are falsely predicted as changing. Both models are often misguided by context changes and large context variety. While high-performances were reached on the SemEval-2020 data, it can not be guaranteed that the performance on other corpora will be as good. It is likely, that at least a small fine-tuning might be necessary to find well-performing parameter configurations, even for SGNS. This requires either gold data or a human annotation process. However, the first is often not available and the latter is time-consuming.

Considering the illustrated strengths and weaknesses, I believe that the framework has a lot of potential and can be helpful to different types of people both in the field of LSCD and outside of it. I also think that the largely automated nature of the framework can be utilized by users with more resources to find generally well-performing parameters and thus eliminating one of its key weaknesses. I hope that the framework will be useful for many people in and outside the field of LSCD, whether they are experienced or not.

References

Jay Allamar. The Illustrated Transformer. `https://jalammar.github.io/illustrated-transformer/`. Accessed: 2021-02-26.

Nikolay Arefyev and Vasily Zhikov. BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. Learning principled bilingual mappings of word embeddings while preserving monolingual invariance. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2289–2294, Austin, Texas, November 2016. Association for Computational Linguistics.

Mikel Artetxe, Gorka Labaka, and Eneko Agirre. A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 789–798, 2018.

Nikhil Bansal, Avrim Blum, and Shuchi Chawla. Correlation clustering. *Machine Learning*, 56(1-3):89–113, 2004. doi: 10.1023/B:MACH.0000033116.57574.95.

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. Overview of the EVALITA 2020 Diachronic Lexical Semantics (DIACR-Ita) Task. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online, 2020. CEUR.org.

Berliner Zeitung. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin, 2018. URL `http://zefys.staatsbibliothek-berlin.de/index.php?id=155`.

Andreas Blank. *Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen*. Niemeyer, Tübingen, 1997.

Deutsches Textarchiv. Grundlage f¨ür ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften, 2017. URL `http://www.deutschestextarchiv.de/`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1423.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. Outta control: Laws of semantic change and inherent biases in word representation models. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1147–1156, Copenhagen, Denmark, 2017.

Romain Futrzynski. Peltarion. `https://peltarion.com/blog/data-science/self-attention-video`. Accessed: 2021-02-26.

Mario Giulianelli, Marco Del Tredici, and Raquel Fernández. Analysing lexical semantic change with contextualised word representations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3960–3973, Online, July 2020. Association for Computational Linguistics.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 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, 2016.

Charles R. Harris, K. Jarrod Millman, Stéfan J van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585:357–362, 2020. doi: 10.1038/s41586-020-2649-2.

Anna Hätty, Dominik Schlechtweg, and Sabine Schulte im Walde. SURel: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the 8th Joint Conference on Lexical and Computational Semantics*, pages 1–8, Minneapolis, MN, USA, 2019.

Anna Hätty, Dominik Schlechtweg, Michael Dorna, and Sabine Schulte im Walde. Predicting degrees of technicality in automatic terminology extraction. In *Proceed-*

ings of the 58th Annual Meeting of the Association for Computational Linguistics, Seattle, Washington, 2020. Association for Computational Linguistics.

Simon Hengchen, Nina Tahmasebi, Dominik Schlechtweg, and Haim Dubossarsky. Challenges for computational lexical semantic change. In Nina Tahmasebi, Lars Borin, Adam Jatowt, Yang Xu, and Simon Hengchen, editors, *Computational Approaches to Semantic Change*, volume Language Variation, chapter 11. Language Science Press, Berlin, 2021. URL `https://arxiv.org/abs/2101.07668v1`.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy: Industrial-strength Natural Language Processing in Python, 2020. URL `https://doi.org/10.5281/zenodo.1212303`.

Renfen Hu, Shen Li, and Shichen Liang. Diachronic sense modeling with deep contextualized word embeddings: An ecological view. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3899–3908, Florence, Italy, 2019. Association for Computational Linguistics.

Jens Kaiser, Dominik Schlechtweg, Sean Papay, and Sabine Schulte im Walde. IMS at SemEval-2020 Task 1: How low can you go? Dimensionality in Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020a. Association for Computational Linguistics. URL `https://arxiv.org/abs/2008.03164`.

Jens Kaiser, Dominik Schlechtweg, and Sabine Schulte im Walde. OP-IMS @ DIACR-Ita: Back to the Roots: SGNS+OP+CD still rocks Semantic Change Detection. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online, 2020b. CEUR.org. URL `https://arxiv.org/abs/2011.03258`. Winning Submission!

Jens Kaiser, Sinan Kurtyigit, Serge Kotchourko, and Dominik Schlechtweg. 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, Online, 2021. Association for Computational Linguistics. URL https://arxiv.org/abs/2101.09368.

Yoon Kim, Yi-I Chiu, Kentaro Hanaki, Darshan Hegde, and Slav Petrov. Temporal analysis of language through neural language models. In *LTCSS@ACL*, pages 61–65. Association for Computational Linguistics, 2014.

Andrey Kutuzov and Mario Giulianelli. UiO-UvA at SemEval-2020 Task 1: Contextualised Embeddings for Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics.

Andrey Kutuzov, Erik Velldal, and Lilja Øvrelid. Tracing armed conflicts with diachronic word embedding models. In *Proceedings of the Events and Stories in the News Workshop*, pages 31–36, Vancouver, Canada, August 2017. Association for Computational Linguistics. doi: 10.18653/v1/W17-2705. URL https://www.aclweb.org/anthology/W17-2705.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 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, 2018. Association for Computational Linguistics.

Severin Laicher, Gioia Baldissin, Enrique Castaneda, Dominik Schlechtweg, and Sabine Schulte im Walde. CL-IMS @ DIACR-Ita: Volente o Nolente: BERT does not outperform SGNS on Semantic Change Detection. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online, 2020. CEUR.org. URL https://arxiv.org/abs/2011.07247.

Severin Laicher, Sinan Kurtyigit, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. Explaining and improving bert performance on lexical semantic change detection. In *Proceedings of the Student Research Workshop at the*

16th Conference of the European Chapter of the Association for Computational Linguistics, Online, 2021. Association for Computational Linguistics.

Omer Levy, Yoav Goldberg, and Ido Dagan. Improving distributional similarity with lessons learned from word embeddings. *Transactions of the Association for Computational Linguistics*, 3:211–225, 2015.

Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In Yoshua Bengio and Yann LeCun, editors, *1st International Conference on Learning Representations, ICLR 2013, Scottsdale, Arizona, USA, May 2-4, 2013, Workshop Track Proceedings*, 2013a.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 2013b.

Neues Deutschland. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin, 2018. URL `http://zefys.staatsbibliothek-berlin.de/index.php?id=156`.

Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 1532–1543, Doha, Qatar, 2014.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2227–2237, New Orleans, LA, USA, 2018.

Martin Pömsl and Roman Lyapin. CIRCE at SemEval-2020 Task 1: Ensembling Context-Free and Context-Dependent Word Representations. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics.

Ondřej Pražák, Pavel Přibáň, and Stephen Taylor. UWB @ DIACR-Ita: Lexical Semantic Change Detection with CCA and Orthogonal Transformation. In Valerio Basile, Danilo Croce, Maria Di Maro, and Lucia C. Passaro, editors, *Proceedings of the 7th evaluation campaign of Natural Language Processing and Speech tools for Italian (EVALITA 2020)*, Online, 2020. CEUR.org.

Radim Řehůřek and Petr Sojka. Software Framework for Topic Modelling with Large Corpora. In *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*, pages 45–50, Valletta, Malta, May 2010. ELRA. `http://is.muni.cz/publication/884893/en`.

Gerard Salton and Michael J McGill. *Introduction to Modern Information Retrieval.* McGraw-Hill Book Company, New York, 1983.

Dominik Schlechtweg and Sabine Schulte im Walde. Simulating lexical semantic change from sense-annotated data. In A. Ravignani, C. Barbieri, M. Martins, M. Flaherty, Y. Jadoul, E. Lattenkamp, H. Little, K. Mudd, and T. Verhoef, editors, *The Evolution of Language: Proceedings of the 13th International Conference (EvoLang13)*, 2020. doi: 10.17617/2.3190925. URL `http://brussels.evolang.org/proceedings/paper.html?nr=9`.

Dominik Schlechtweg, Sabine Schulte im Walde, and Stefanie Eckmann. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana, 2018.

Dominik Schlechtweg, Anna Hätty, Marco del Tredici, and Sabine Schulte im Walde. 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, 2019. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. SemEval-2020 task 1: Unsupervised Lexical Semantic

Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. submitted.

Hinrich Schütze. Automatic word sense discrimination. *Computational Linguistics*, 24(1):97–123, March 1998.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. Survey of Computational Approaches to Diachronic Conceptual Change. *arXiv e-prints*, 2018.

Hiroya Takamura, Ryo Nagata, and Yoshifumi Kawasaki. Analyzing semantic change in Japanese loanwords. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1195–1204, Valencia, Spain, 2017. Association for Computational Linguistics.

Peter D. Turney and Patrick Pantel. From frequency to meaning: Vector space models of semantics. *J. Artif. Int. Res.*, 37(1):141–188, January 2010. ISSN 1076-9757.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Lilian Weng. Learning word embedding. *lilianweng.github.io/lil-log*, 2017. URL `https://lilianweng.github.io/lil-log/2017/10/15/learning-word-embedding.html`.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander M. Rush. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online, October 2020. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/2020.emnlp-demos.6`.

George K. Zipf. *Human Behaviour and the Principle of Least Effort.* Addison-Wesley, 1949.

George Kingsley Zipf. *The Psychobiology of Language.* Houghton-Mifflin, New York, NY, USA, 1935.

A German Summary

A.1 Einleitung

Im Rahmen dieser Thesis wird ein Framework zur vollautomatisierten Entdeckung von Wörtern die einen Bedeutungswandel durchgemacht haben bereitgestellt.

Wörter verändern ihre Bedeutung mit der Zeit. Bedeutungen können verloren gehen, aber es können auch neue dazugewonnen werden. Das Gebiet LSCD (Lexical Semantic Change Detection) beschäftigt sich mit der automatisierten (mit Hilfe von Computern) Erkennung solcher Bedeutungswandel. Durch extensive Forschung existieren nun sowohl optimierte Modelle, als auch hoch qualitative Datensätze für verschiedene Sprachen, die zur Analyse, Auswertung und Verbesserung der Modelle genutzt werden können. Allerdings liegt der Fokus nur selten auf der Entdeckung neuer Wörter die einen Bedetungswandel durchgemacht haben. Das Ziel dieser Thesis ist es einen praktischen Nutzen aus den Meilensteinen der bisherigen Forschung zu ziehen.

Dafür wird ein Framework bereitgestellt, dass gängige Methoden nutzt, um vollautomatisiert neue Wörter zu entdecken, die einen Bedetungswandel durchgemacht haben. Zusätzlich werden Hilfsmittel zur Analyse und Optimierung der implementierten Modelle bereitgestellt. Das Framework soll für möglichst viele Nutzer hilfreich sein, unabhängig davon wie viel Erfahrung sie im Gebiet haben. Durch den hohen Automatisierungsgrad könnten auch Nutzer außerhalb des Gebietes vom Framework profitieren.

A.2 Das Framework

Das Hauptaufgabe des Framework's ist die vollautomatisierte Entdeckung von Wörtern die einen Bedeutungswandel durchgemacht haben. Ausgangspunkt dafür ist ein Textkorpuspaar (K_1, K_2). Mit Hilfe des Framework sollen nun Wörter aus dem Durchschnitt der Korpusvokabulare entdeckt werden die einen Bedeutungswandel durchgemacht habe. Außerdem, kann das Framework auch genutzt werden um die beiden verwandten Teilaufgaben (binary classification und graded ranking) aus dem SemEval-2020 shared task (Schlechtweg et al., 2020) vollautomatisiert zu lösen.

Zur Entdeckung von Wörtern die einen Bedeutungswandel durchgemacht haben, werden die folgenden Schritte ausgeführt:

1. Ein neuronales Sprachenmodell (SGNS, BERT) wird verwendet um Worteinbettungen (Darstellung von Wörtern in From eines numerischen Vektors) für Wörter aus dem Durschnitt der Korpusvokabulare zu generieren.

2. Unterschiede zwischen Worteinbettungen aus K_1 und Worteinbettungen aus K_2 werden gemessen.

3. Ein Schwellenwert wird in Abhängigkeit dieser Unterschiede berechnet. Wörter deren Wert höher als dieser Schwellenwert ist, werden als Wörter markiert die einen Bedeutungswandel durchgemacht haben.

4. Unerwünschte Worte (wie z.B., Eigennamen und fremdsprachige Wörter) werden ausgefiltert.

5. (Optional) Die Verwendungen der markierten Wörter werden extrahiert und in einem spezifischen Format gespeichert. Diese können dann genutzt werden um die Güte zu evaluieren oder fälschlicherweiße markierte Entdeckungen zu finden.

Das Framework bietet zur Generierung der Worteinbettungen ein statisches und ein kontextbezogenes Modell an. Der Unterschied ist, dass statische Modelle eine Worteinbettung pro Wort generieren. Kontextbezogene Modelle hingegen, generieren eine Worteinbettung pro Wortverwendung (Satz in dem das Wort vorkommt). Der Entdeckungsprozess unterscheidet sich leicht in abhängigkeit vom gewählten Modell.

Statisches Modell Das Skip-gram with Negative Sampling Modell (SGNS, Mikolov et al., 2013a;b) wird genutzt um je einen Vektorraum für K_1 und K_2 zu generieren. Diese enthalten dann die Worteinbettungen für die entsprechenden Wörter aus K_1

und K_2. Zur Optimierung werden die Worteinbettungen dann normiert und zentriert. Anschließend werden die beiden Vektorräume angeglichen (Orthogonal Procrustes). Für jedes Wort aus dem Durschnitt der Korpusvokabulare wird dann mit Hilfe der Cosine Distance (CD), der Unterschied zwischen der Worteinbettung aus K_1 und der Worteinbettung aus K_2 gemessen.

Kontextbezogenes Modell Kontextbezogene Modelle wie Bidirectinal Encoder Representations from Transformers (BERT, Devlin et al., 2019) benötigen Wortverwendungen um die Worteinbettungen zu generieren. Der Rechenaufwand für die Extrahierung der Verwendungen und die darauf folgende Generierung der Worteinbettungen steigt erheblich mit der Anzahl an Wörtern. Um diesen Aufwand zu verringern, wird ein Skript bereitgestellt, dass eine Stichprobe aus dem Durchschnitt der Korpusvokabulare zieht. Die Größe der Stichprobe kann durch einen Parameter im Skript bestimmt werden. Danach werden mit Hilfe des Modells, für jedes Wort aus der Stichprobe, zwei Mengen von Worteinbettungen generiert. Der Unterschied zwischen zwei Mengen kann mit Hilfe der Average Pairwise Distance (APD) oder der Cosine Similarity (COS) gemessen werden.

Schwellenwertbildung In Abhängigkeit der gemessenen Unterschiede wird ein Schwellenwert berechnet. Dieser entspricht

$$TH = \mu + t \cdot \sigma, \tag{12}$$

wobei μ der Mittelwert und σ die Standardabweichung ist. Wörter mit einem gemessenen Unterschied größer oder gleich diesem Schwellenwert, werden dann als Wörter die einen Bedeutungswandel durchgemacht haben markiert.

Filterung Zwei Filter werden angewandt um unerwünschte Wörter auszusortieren:

1. Den ersten Filter können nur Nomen, Verben und Adjektive passieren.
2. Der zweite Filter entfernt Wörter, bei denen mindestens 10% der Verwendungen entweder fremdsprachig sind, oder mehr als 25% Interpunktion enthalten.

Speicherung für die Annotation In einem optionalen Schritt können Verwendungen, für die markierten Wörter die nicht ausgefiltert wurden, extrahiert und in einem spezifischen Format gespeichert werden. Diese können dann direkt in das DURel Annotationssystem hochgeladen werden. Mit Hilfe des Annotationssystems kann die Güte der Entdeckungen evaluiert werden. Außerdem können fälschlicherweiße markierte Wörter entdeckt und entfernt werden.

A.3 Exemplarische Anwendung

Das Framework und seine Qualität wird anhand eines exemplarischen Entdeckungsprozesses illustriert. Der Prozess beinhaltet mehrere Optimierungsphasen um hochperformante Parameterkonfigurationen zu erhalten. Dafür wird erst der Subtask 2 (graded ranking) gelöst. Die besten Resultate werden dann verwendet um den Subtask 1 (binary classification) zu lösen. Für beide Modell wird dann jeweils die Parameterkonfiguration die beim Subtask 1 am besten abschneidet für die eigentliche Aufgabe der Entdeckung verwendet. Die Güte wird im Anschluss mit Hilfe des DURel Annotationssystem evaluiert.

Resultate Beiden Modelle erreichen ähnlich gute Resultate, für beide Subtasks. Die optimierten Parameterkonfigurationen beider Modelle eignen sich gut um Wörter zu entdecken die einen Bedeutungswandel durchgemacht haben. Allerdings schneidet das statische Modell deutlich besser ab. Bei 19 (70%) der 27 markierten Wörter stimmen die Annotationen von Menschenhand mit den Modellvorhersagen überein. Im kontextbezogenen Fall sind es nur 17 (56%) von 30. Das Wort *Zehner* und das Wort *Sprachrohr* sind zwei der vielen Wörter die im Entdeckungsprozess gefunden wurden und tatsächlich einen Bedeutungswandel durchgemacht haben.

Fälschlicherweiße markierte Wörter und deren Verwendungen werden genauer untersucht um mögliche Fehlerquellen ausfinding zu mache. Die meisten falschen Entdeckungen können dabei einer der folgenden Fehlerquellen zugeordnet werden:

1. **Kontextwandel**: Die erste Kategorie beinhaltet Wörter, deren Kontext zwischen K_1 und K_2 einen Wandel durchmacht. Beide Modelle entdecken diesen

Wandel. Allerdings hat das entsprechende Wort keinen Bedeutungswandel durchgemacht. Beispiele hierfür sind *Angriffswaffe*, *Ächtung* und *Königreich.*

2. **Kontextvariation**: Die zweite Kategorie beinhaltet Wörter die in vielen unterschiedlichen Kontexten eingesetzt werden können. In diesen Fällen, werden beide Modelle durch die starke Kontextvariation hinters Licht geführt. Beispiele enthalten unter anderem, *neunjährig*, *vorjährig* und *Bemerken.*

A.4 Fazit

Alles in allem zeigt der exemplarische Prozess, dass das Framework erfolgreich zur Entdeckung von Wörtern die einen Bedeutungswandel durchgemacht haben genutzt werden kann. Das statische Modell funktioniert dabei allerdings deutlich besser. Außerdem wurde das Framework zum Lösen der beiden Subtasks aus dem SemEval-2020 shared task genutzt. Auch hier konnten beide Modelle gute Resultate erzielen.

Es wurden allerdings auch Schwachstellen des Frameworks verdeutlicht. Neben den Wörtern die tatsächlich einen Bedeutungswandel durchgemacht haben, werden von beiden Modellen auch solche markiert, die keinen Bedeutungswandel durchgemacht habe. Zusätzlich werden andere gar nicht erst entdeckt. Außerdem kann nicht garantiert werden, dass sich die hohe Performanz die auf den SemEval Datensatz erreicht wurde, auf andere Datensätze überträgt. Ein kleine Optimierungsphase wird vermutlich nötig sein, auch für SGNS. Dies benötigt allerdings Golddaten oder eine Annotation von Menschenhand. Ersteres ist selten vorhanden, letzteres sehr aufwendig.

Erklärung

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Druck-Exemplaren überein.

Datum und Unterschrift: 26.02.21 S.Kurtyigit

Declaration

I hereby declare that the work presented in this thesis is entirely my own. I did not use any other sources and references that the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted hard copies.

Date and Signature: 26.02.21 S.Kurtyigit