

Modelling Semantic Change on Twitch.tv Chat Messages

Master's Thesis

submitted by

Armin Bernstetter

August 31st 2021



Julius-Maximilians-Universität Würzburg

Lehrstuhl für Informatik X

Data Science

Supervisor:

M.Sc. Albin Zehe

Examiners:

Prof. Dr. Andreas Hotho

Prof. Dr. Frank Puppe

Erklärung

Ich erkläre hiermit, dass ich die vorliegende Masterarbeit selbständig und ohne unzulässige, fremde Hilfe verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe.

Die bildlichen Darstellungen, Tabellen, Quelltexte und Graphen habe ich, sofern nicht ausdrücklich anders angegeben, selbst angefertigt.

Die Arbeit habe ich bisher oder gleichzeitig keiner anderen Prüfungsbehörde vorgelegt.

Würzburg, den

We must not let our desires for a specific result cloud our perceptions.

But how can we not, in searching, wish for a specific result? What scientist goes into a project without a hope for what they will find?

- Brandon Sanderson,
The Stormlight Archive Book 4 - Rhythm of War

Abstract

Lexical semantic change detection is an area in the field of natural language processing that researches the shift in meaning and usage of words over time or over different domains. Semantic change detection has previously mostly been applied to historic texts, however, recently data e.g. from social media such as *Twitter* has been used.

The live-streaming platform *Twitch.tv* is one of the most popular websites for live entertainment content, especially in the area of gaming. Live content, often produced by solo entertainers which are called streamers, is displayed in channels. Each of these channels has a live chatroom where viewers can post messages. These messages are often very short, shorter even than e.g. tweets on *Twitter* and often contain so-called *emotes*. These emotes are unique to the platform and different from unicode emojis used e.g. in messenger applications. A novel text-based data set is created from Twitch chat message and used in this work.

This thesis reproduces results of previous work in this area and applies these methods to the novel dataset. It is explored whether one year of data gathered from *Twitch.tv* chat messages contains noticeable semantic change and examines that especially the Covid pandemic has an influence on the meaning of certain words. Additionally, domain-specific semantic change in the context of two video games is successfully examined. A method shifting the focus of semantic change detection on *Twitch.tv* more towards utilizing emotes is designed and implemented. This method treats chat messages as multimodal data with emotes as additional modality next to plain text. The results of this experimental approach could not achieve competitive performance compared to the results provided by established methods.

Zusammenfassung

Sprachwandel, oder Semantic Change und dessen automatische Erkennung ist ein Forschungsbereich im Feld des Natural Language Processing. Dieser Forschungsbereich untersucht den Bedeutungs- oder Benutzungswandel, welchen Wörter über Zeit durchlaufen. Ebenfalls beinhaltet dies die Untersuchung von Bedeutungsunterschieden von Wörtern zwischen unterschiedlichen Textarten bzw -domänen. In der Vergangenheit wurde die Erkennung von Sprachwandel vor allem auf historischen Texten angewandt. In jüngerer Zeit wurde jedoch auch Forschung betrieben mit Textdaten beispielsweise der Plattform *Twitter*.

Die live-streaming Plattform *Twitch.tv* ist eine der beliebtesten Webseiten für Unterhaltungsinhalt, besonders im Bereich Gaming. Liveinhalte, häufig produziert von Alleinunterhalter:innen - so genannten Streamer:innen - wird über deren Kanäle ausgestrahlt. Jeder dieser Kanäle beinhaltet einen Chatraum, in welchem Zuschauer:innen die Möglichkeit haben, Nachrichten zu schreiben. Häufig sind diese Nachrichten sehr kurz, oftmals sogar kürzer als Tweets auf der Plattform *Twitter*. Ebenfalls beinhalten Nachrichten auf Twitch häufig so genannte *Emotes*. Diese Emotes sind einzigartig auf Twitch und unterscheiden sich von normalen Unicode Emojis wie sie beispielsweise in Messenger Apps benutzt werden. In dieser Arbeit wird ein neuartiger Textdatensatz aus Twitch-Chatnachrichten benutzt.

Diese Arbeit reproduziert Ergebnisse vorheriger Arbeiten in diesem Bereich und wendet diese Methoden auf den neuen Twitch-Datensatz an. Es wird erarbeitet, ob bereits in Textdaten, welche Twitch-Nachrichten aus nur einem Jahr beinhalten, erkennbaren Sprachwandel zeigen. Hier kann besonders die Covid-19 Pandemie als maßgebende Ursache für den Bedeutungswandel einiger Wörter herausgearbeitet werden. Ebenso wird domänenspezifischer Sprachwandel im Kontext spezifischer Videospiele erfolgreich untersucht. Eine neue Methode wird entwickelt, welche den Fokus der automatischen Erkennung von Sprachwandel auf Twitch verschiebt in Richtung der Emotes. Diese Methode behandelt Twitch-Chatnachrichten als multimodale Daten, mit Emotes als separate Modalität zusätzlich zu den reinen Textdaten. Die Ergebnisse dieses Ansatzes erreichen keinen vergleichbar guten Erfolg wie die Ergebnisse welche durch etablierte Methoden erreicht werden.

Contents

Erklärung	iii
I. Thesis	1
1. Introduction	3
2. Background	7
2.1. Twitch.tv	7
2.1.1. Overview	7
2.1.2. History	8
2.1.3. Twitch’s Chat Language	9
2.2. Lexical Semantic Change	10
2.2.1. Diachronic and Synchronic Semantic Change	10
2.2.2. Types of Semantic Change	11
2.3. Multimodality	13
2.4. Technical Background	14
2.4.1. Neural Networks	14
2.4.2. Neural Word Embeddings	14
3. Related Work	17
3.1. Semantic Change Detection	17
3.2. Twitch, Emotes, and Emoji	19
3.3. Multimodal Deep Learning	19
4. Methods for Lexical Semantic Change Detection	21
4.1. Semantic Change Detection	21
4.1.1. Two-step Approach	21
4.1.2. Change-point detection	23
4.2. Multimodal Semantic Change Detection	24
4.2.1. Plan/Architecture	25
4.2.2. Vocabularies/Combinations	25
4.2.3. Emote Representations	28
4.2.4. Fusion	28
5. Data	31
5.1. Datasets for Reproduction Experiments	31
5.1.1. Deutsches Textarchiv (DTA)	31

5.1.2.	DURel and SURel	31
5.2.	Synthetic Dataset	32
5.2.1.	Synthetic Dataset Generation Framework	33
5.2.2.	Synthetic Datasets in this Work	33
5.3.	Twitch Data Analysis	34
5.4.	Twitch Preprocessing	34
5.4.1.	Extracted Information	34
5.4.2.	Message Grouping	37
5.4.3.	Emote Corpus	37
5.4.4.	Games Corpora	37
5.4.5.	Dataset for the Multimodal Approach	38
6.	Experiments and Results	39
6.1.	Word Embeddings	39
6.2.	Reproducing Wind of Change	40
6.3.	Change-point Analysis of German Texts	42
6.4.	Synthetic Dataset Experiments	45
6.4.1.	Evaluation Method	45
6.4.2.	Synthetic Twitch Dataset: Comparing Different Message Grouping Setups	46
6.4.3.	Comparing Synthetic Results: Twitch vs Twitter vs DTA	51
6.5.	Monitoring Selected Words with presumed LSC	52
6.5.1.	The Chosen Words and their Frequencies	52
6.5.2.	Change-point Detection of Chosen Words	55
6.6.	Lexical Semantic Change in Games	56
6.6.1.	Dota 2 vs. League of Legends	57
6.6.2.	Game-specific Time Series - Dota 2	57
6.7.	Multimodal Semantic Change	61
6.7.1.	Synthetic Dataset Baseline	61
6.7.2.	Emotes as Words	61
6.7.3.	Emotes as Images	61
6.7.4.	Vocabulary with Emotes vs Vocabulary without Emotes	62
6.7.5.	Configuration numbers	62
6.7.6.	Results	62
7.	Discussion and Future Work	65
8.	Conclusion	69
	Bibliography	76

II. Appendix	83
A. Appendix	85
A.1. Embedding Parameters	85
A.2. Dota 2 and League of Legends Domain Specific Usage Changes	86

Part I.

Thesis

1. Introduction

This thesis investigates semantic change on a previously unexplored area, comments on the popular streaming platform *Twitch.tv*².

Motivation Language is not a fixed construct. At least when it comes to languages that are still in use. Noam Chomsky once said:

“Language is a process of free creation; its laws and principles are fixed, but the manner in which the principles of generation are used is free and infinitely varied. Even the interpretation and use of words involves a process of free creation.”

Over time, language is subject to influences and changes. American or Australian English use different words than British English, many European languages share a common ancestor in Latin but have diverged over centuries. Language adapts to the current time, as new inventions or societal evolutions beget changes with new words or new word usages. Language dynamically adapts e.g. to changing societal norms, getting ever more inclusive, redefining words or adding new meanings to words.

With the rising popularity of products by the company *Apple*, the word which once only had the meaning of the fruit, gained another meaning as the proper noun signifying the company. Another example is the word *gay* whose semantic trajectory from the context of “cheerful” to an LGBTQ+ context can be seen in figure 1.1.

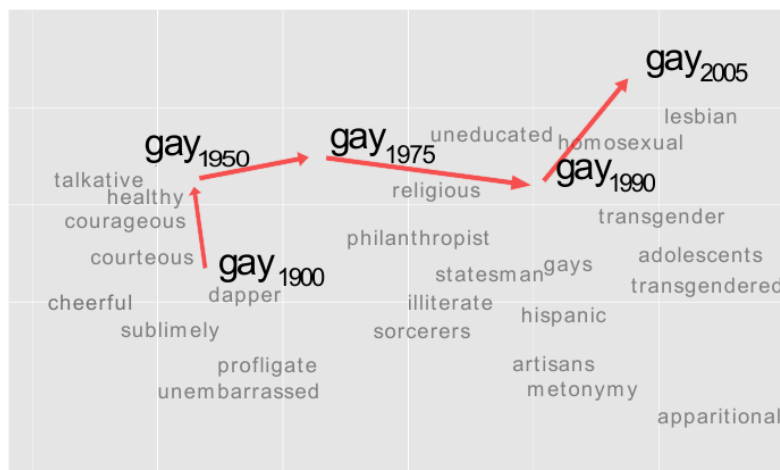


Figure 1.1.: This figure illustrates the change of meaning which the word *gay* went through over the last century (Kulkarni et al., 2015).

²<https://www.twitch.tv/>

1. Introduction

These changes in meaning are called *lexical semantic change* and have been subject of linguistic research for a long time. More recently, semantic change detection has been researched in natural language processing (NLP) with computer scientists using various text mining techniques to detect changes in word meanings. Traditionally, semantic change research has required knowledge of domain experts as well as datasets spanning at least multiple decades if not even centuries. Modern NLP techniques offer many new possibilities such as automatic detection of semantic change or detecting change which domain experts might not even have noticed.

With the increasing usage of the internet in recent decades, not only traditionally used texts such as novels or articles have provided data for NLP research but also posts from social media platforms. Micro-blogging services such as Twitter have been very popular in areas such as sentiment analysis but also semantic change detection. Meaning changes in social media can happen incredibly quickly with an unstopping influx of new trends and memes.

A less researched platform is *Twitch.tv*, a live streaming site mostly for gaming-related content. Users called *streamers* can broadcast whatever content they want on their channel under multiple categories. Most of the content is focused on broadcasting gameplay but there's also musicians and artists streaming themselves creating art or streamers just chatting with their viewer. Additionally, events such as esports tournaments and - especially in times of Covid - expo events such as the E3, GamesCom or press conferences by companies, are broadcast on Twitch.

Viewers can comment on these live streams in a live chat constantly running at the side of a stream. Most of the messages posted in these chats are rather short, especially on popular channels with high viewer counts. A characteristic which is unique to Twitch is the usage of *Emotes* which are unlike the widely used unicode emoji. Each of these emotes has a text representation which users type in the chat, resulting in the emote appearing as image. Figure 1.2 shows a screenshot of such a comment section on Twitch. As an indirect precursor of this work, Kobs et al. (2020) have found that messages on Twitch and especially emotes, some of which are visible in the screenshot, carry enough meaning to provide results for sentiment analysis. Sentiment analysis is a research area in the field of natural language processing with the goal of automatically detecting the sentiment or emotion of a given text. Similar to other social media platforms, Twitch is filled with memes and inside jokes which often have a very short half life. This might lead to semantic change happening rather quickly, in contrast to traditional historic semantic change which

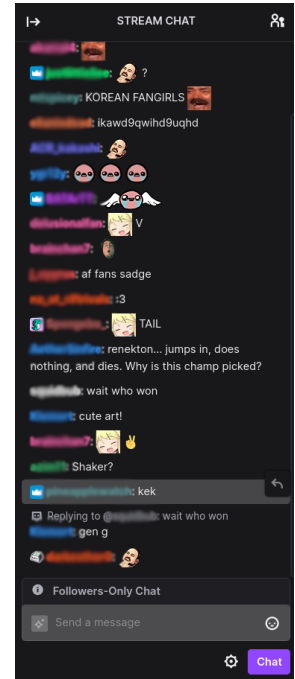


Figure 1.2.: Screenshot of a Twitch comment section. The language used in the comments is fairly different from common English. In the bottom right, an emote picker helps with selecting an emote. User names are anonymized due to privacy reasons. This chat is in *followers-only* mode which means only users who are following the channel can post a message.

happens over decades or even centuries. This begets the question whether semantic change can be detected in Twitch chat messages, a question which to the author’s knowledge previously has not been explored.

This thesis explores the validity of established semantic change detection methods by reproducing previous results. Then, these methods are applied to Twitch chat data. Afterwards, an expanded multi-modal method adapted to the emote-heavy texts on Twitch is developed. This new method sees emotes as a secondary modality next to the *text* modality.

Thesis Structure Chapter 2 gives an overview of the platform Twitch, describes general terms and background of semantic change detection, introduces the concept of multimodality, and introduces some technical basics. Chapter 3 explores related work in the general area of semantic change as well related work on Twitch and other social media. Chapter 4 introduces the technical methods and algorithms used for semantic change detection and Chapter 5 describes the datasets used in this work. Chapter 6 presents the detailed settings and experiments conducted in this work as well as their results. Chapter 7 discusses the results and content of this work and proposes some areas of future work. Finally, Chapter 8 concludes this thesis.

2. Background

This chapter is aimed at introducing the basics necessary for the remainder of this work. Section 2.1 gives background information on Twitch and its peculiarities. Section 2.2 introduces the field of semantic change detection and Section 2.3 the concept of multi-modality in machine learning and natural language processing. Finally, Section 2.4 gives a brief overview of neural networks as well as word embeddings which are extensively used in this work.

2.1. Twitch.tv

Twitch is a popular online live-streaming platform and is one of the most popular sites when it comes to live streaming like YouTube is one of the most popular platforms for “VOD” (i.e. video on demand) content. This section gives an overview of Twitch, its background, and its intricacies. It introduces Twitch’s history, which kinds of content are available via Twitch, and its users. These users write the chat messages on Twitch’s website which are the basis of this work’s data set.

2.1.1. Overview

Twitch.tv is a live streaming platform on which individuals (so called “streamers”) or companies can broadcast live content. These live streams are shown on users’ profile pages which are called “channels”. Past broadcasts are saved as VODs on a channel’s page, available for users to watch even when not live. Similar to many other platforms online, viewers can “follow” channels. This allows them to have an easily available overview of which of their followed channels is currently broadcasting live. Additionally, users can “subscribe” to a channel for a monthly fee to support a streamer monetarily. This is available once streamers are “partnered” with the company Twitch. Part of this subscription fee goes to Twitch and part of it supports the streamer. Subscribing to a channel provides access to channel-specific emotes which can be used anywhere on Twitch, even in chat channels of other streamers e.g. to show that a user is subscribed to another channel.

Every channel has a “Stream Chat” in which viewers can write comments during the live stream. A replay of this chat is available for VODs and viewers can also comment on these videos later at the currently viewed timestamp as if they had written the comment during the live stream. These stream chats can have several configuration modes which a channel’s “moderators” can set. For example, a chat can be put into “subscriber-only”, “follower-only”, “emote-only”, or “slow” mode. The former two are self explanatory, the latter two are modes where users can only post emotes but no other text, or can only

2. Background

post messages every X seconds respectively. Moderators of a channel have powers which allow them to e.g. delete messages, time-out or ban users who disregard Twitch's terms of service or the "netiquette" set by a streamer.

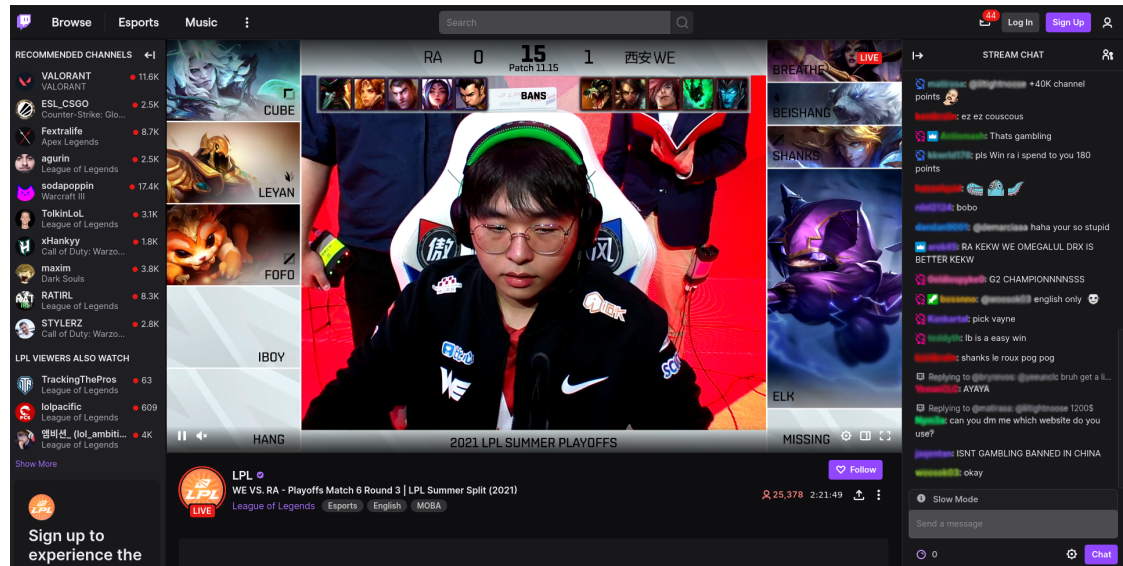


Figure 2.1.: A screenshot of a Twitch channel currently live streaming a League of Legends Esport tournament. The current live stream is visible in the middle of the screen. To the left is a column where - if a user is logged in - the currently streaming, followed channels are listed. Without logging in, channels recommended by Twitch are listed. To the right, the stream chat window is visible.

2.1.2. History

Twitch was founded in 2011 as a gaming-related side project of the streaming platform Justin.tv¹. Initially, Twitch was meant as platform to broadcast competitive esport but soon received more popularity than Justin.tv leading to Twitch becoming the company's main focus². In 2014, Twitch was acquired by Amazon³. Twitch's content is still mostly gaming-related with streamers broadcasting themselves playing videogames, esports competitions, and events such as press conferences related to gaming. Especially in times of Covid-19, Twitch has been a useful platform to be able to hold certain events live and online, e.g. the E3 Expo 2021⁴. Streamers have also expanded to more non-gaming content in recent years with music or cooking broadcasts or simply talking to or with their viewers in so called "Just Chatting" streams. Similar to YouTube where channels

¹<https://www.businesswire.com/news/home/20110606005437/en/Justin.tv-Launches-TwitchTV-World%E2%80%99s-Largest-Competitive-Video> (last accessed 2021-08-17)

²<https://www.theverge.com/2014/8/5/5971939/justin-tv-the-live-video-pioneer-that-birthed-twitch-officially-shuts> (last accessed 2021-08-17)

³<https://blog.twitch.tv/en/2014/08/25/a-letter-from-the-ceo-august-25-2014-b34c1cfbb099/>




⁴https://en.wikipedia.org/wiki/E3_2021 (last accessed 2021-08-17)




can monetize their content, Twitch has also become a source of income for many of the more popular streamers. Advertisement revenue, subscription money, and donations by viewers can amount to high earning numbers. For example, the streamer *Pestily* who - as of July 2021 according to Twitchstats¹ - had the highest subscriber count with a total of around 80,000 users, which amounts to at least 400,000\$ paid by viewers per month. Depending on their contract with Twitch, half or more of that amount is earned by the streamer. Additional to this, streamers often have product placements or other advertisement deals with companies, same as influencers on other platforms.

2.1.3. Twitch's Chat Language

The messages posted in a stream chat on Twitch are very different from common English and often shaped by the usage of memes, inside jokes, and Twitch emotes. Emotes are little pictures - sometimes animated - which can be added by users to their messages in the chat. They play an essential part in the communication in a stream chat as they allow users to express meaning in a much more concise way compared to writing verbose texts. They mostly replace other forms of emoticons such as Unicode emoji on Twitch. Image 1.2 from Chapter 1 showed a small excerpt from a stream chat including heavy emote usage. Twitch emotes can be added to a message in two different ways. On one hand, there is an emote picker next to the chat message form from which a user can click any emote that is available to them. On the other hand, each emote has a text representation and a user can simply write this name in the chat and Twitch will render it as the according image. If an emote is available for a user, the image will render. If a user e.g. tries to invoke a subscriber-emote of a channel they are not subscribed to, only the emote name will show in the chat instead of the image.

Emotes have their own specific meanings, back stories and are all used in more or less different ways. Some of those meanings can be guessed by simply looking at the emote image. Others are not so clear at first glance and are based on internet memes or inside jokes

One of the most popular emotes on Twitch is and has been for a long time, the emote  Kappa . This emote has evolved to denote sarcasm when appended to a message². The sentence “Good game!  Kappa ” could be an example where the commenter makes fun of the streamer by congratulating them sarcastically. Emotes like  Kappa can therefore be modifiers for the meaning of a message.

Emotes in this thesis will be represented as above, combining both the emote image and the emote text representation (e.g.  Kappa ,  DansGame ,  4Head). This, however, is not how emotes are represented in the Twitch chat. There, only the image is visible but not the text.

Twitch itself currently has around 280 “global” emotes³ which can be used by any user in any channel. Additionally, there is a very large number of subscriber emotes. Each channel can publish a varying number of emotes depending on their subscriber

¹<https://twitchstats.net/real-sub-count/2021/July>(last accessed 2021-08-17)

²<https://knowyourmeme.com/memes/kappa> (last accessed 2021-08-17)

³<https://twitchemotes.com/> (last accessed 2021-08-17)

2. Background

count with very popular channels having more emote-slots granted by Twitch¹. Another popular source of emotes are (browser) extensions such as “Better Twitch TV” (BTTV)² or “FrankerFaceZ” (FFZ)³. These applications allow users to create own emotes and add them to the database from which streamers can then add those emotes to their channels independently of Twitch’s own emote limits. These extension-emotes, however, can only be seen by other users who have installed the extension. In a survey conducted among Twitch users, Kobs et al. (2020) found that around 60% of the participants were using at least one of these extensions.

2.2. Lexical Semantic Change

The study of semantic change has been an area of interest in linguistics and especially historical linguistics for a long time. Williams (1976) mentions a “*century-old failure of historical linguistics to discover regularities of semantic change comparable to those in phonological change*”. As one reason for that he brings up the question of what “meaning of a word” even means, i.e. what someone “[wants] to know when [they] ask what a word means”. Lichtenberk (1991) notes that meanings of linguistic elements are subjective and not fully determined but rather open-ended and flexible.

“This is what allows words to be applied to new experiences, to express newly perceived relations among phenomena and thus to form new categories or to alter the make-up of existing categories, and to relate to each other nomena from different cognitive domains.” (Lichtenberk, 1991)

In recent years, semantic change has been recognized as an area which can benefit from the advancements made in the field of natural language processing and computational linguistics. The utilization of NLP methods beckons with the ability to automatically detect semantic change without having to rely solely on the theoretical meaning of words. Automatically examining usage changes e.g. by looking at the neighborhood/context of a word in different time periods produces potential candidates of words which might have undergone changes in meaning or usage.

2.2.1. Diachronic and Synchronic Semantic Change

A classic approach to lexical semantic change detection (LSCD) is to take texts from different points in time and compare word usages. In linguistics, such an approach is called diachronic from greek *dia* “through” and *chronos* “time”. In contrast, a synchronic approach takes into consideration texts from the same time period but not necessarily the same domain. This can mean comparing texts e.g. from different dialects (British, American, Australian etc.) but also texts from varying text domains such as different genres of literature. For example a “virus” in medical texts has a different meaning than in texts related to IT security.

¹https://help.twitch.tv/s/article/subscriber-emote-guide?language=en_US (last accessed 2021-08-17)

²<https://betterttv.com/> (last accessed 2021-08-17)

³<https://www.frankerfacez.com/> (last accessed 2021-08-17)

2.2.2. Types of Semantic Change

Semantic change can happen in different ways. Words can gain an additional meaning or can be used in an additional sense (e.g. the word “Apple” gaining the usage as a proper noun meaning the tech company). They can lose one sense but gain another context such as the word “gay” (see Figure 1.1), they can completely lose a sense without replacement, or fall out of use entirely.

Innovative Meaning Change

This type of semantic change, called “Innovative Meaning Change” by Koch (2016, p. 24) in reference to Blank (1997), covers the case of a word gaining an additional sense. Initially a word has one meaning, then a new sense emerges and gains in popularity, resulting in polysemy of equal status between the two senses. According to Koch, polysemy is then the synchronic result of this innovative meaning change. A simple example would be the German word “Steckenpferd” which initially had only the literal sense of a wooden horse (“Pferd”) on a stick (“Stecken”) i.e. a child’s toy. Over time, this word gained the meaning of “hobby” or “favorite activity”. This polysemous state can actually endure for a long time. In the example of “Steckenpferd” already the Brothers Grimm in their dictionary of the German language in the 19th century reported the meaning of “hobby”¹. Another example here again is the word “apple” which gained the sense of the proper noun meaning the tech company.

Reductive Meaning Change

The flipside of innovative meaning change is reductive meaning change (Koch, 2016, p. 26). With a word starting out with polysemous meanings, a sense slowly falls out of use resulting in the loss of a sense. Again, the example of the word “gay” can be used here. Any other sense has fallen out of use over the last decades until only the context of “homosexuality” remained.

Cycle of Genesis and Disappearance

Together, innovative and reductive meaning change form a “cycle of genesis and disappearance” (Koch, 2016, Fig. 2). A word gains an additional sense over time, spends a certain time in a polysemous state after which the older sense falls out of use again. Koch (2016) mentions that often in literature, only this full cycle is considered as *a* meaning change as only at the end of the cycle, the meaning of a word has completely changed. He argues against this by stating that both innovative and reductive meaning change can happen as mutually independent processes.

¹<https://woerterbuchnetz.de/?sigle=DWB&lemma=Steckenpferd> (last accessed 2021-08-17)

2. Background

Further Segmented Change Types

Other literature mentions more fine-grained types of semantic change. In their survey on literature on semantic change detection Tahmasebi et al. (2018) gather types of semantic change that are examined in the surveyed literature. Table 2.1, taken from Tahmasebi et al. (2018, p. 35) shows the accumulation of these change types. “Novel word sense” for example is what Koch calls “innovative meaning change”.

Change Type	Description
Novel word	a new word with a new sense, or for that word new sense (any neologism such as <i>Internet</i>)
Novel word sense	a novel word sense that is attached to an existing word (e.g. <i>virus</i> with its new meaning of computer virus)
Novel related word sense	a novel word sense that is related to an existing sense.
Novel unrelated word sense	a novel word sense that is unrelated to any existing sense. (most of the cases where a word suddenly gets used as proper noun e.g. <i>Apple</i>)
Broadening	a word sense that is broader in meaning at a later time (again for example <i>virus</i> or <i>surfing</i> in their IT senses.)
Join	two word senses that exist individually and then join at a later time
Narrowing	a word sense that is broader in meaning at an earlier time
Split	a word sense that splits into two individual senses at a later time
Death	a word sense that is no longer used (e.g. <i>awful</i> was originally positively connotated but is now only used negatively)
Change	any significant change in sense that subsumes all previous categories

Table 2.1.: Table taken from Tahmasebi et al. (2018, p. 35). It shows a more fine-grained segmentation of different change types which are investigated in the literature surveyed in Tahmasebi et al.’s paper. These categories cannot always be separated cleanly.

2.3. Multimodality

One of the goals of this work is to use the multimodal nature of Twitch comments, consisting of text and images (emotes), to improve semantic change detection in this domain. Information in the real world generally comes in multiple channels or modalities such as text, audio, or images. Multimodality as a concept in everyday life as well as in research can apply to many areas, sometimes very different from each other. In natural phenomena, it is rare that one modality provides complete knowledge of the phenomenon (Lahat et al., 2015). One of the most intuitively understood examples is audio-visual multimodality. From film to web videos, this type of multimodality is all around us. Other examples where the term *multimodality* can apply as well, could be in the medical area. Monitoring the status of a patient e.g. the brain activity can be done by using electroencephalography (EEG) as well as magnetic resonance imaging (MRI). In environmental studies such as meteorological monitoring and earth observation as well, data can be multimodal. Many different sensors can report on various activities such as rainfall, temperature, atmospheric pressure etc.

In this work, however, the multimodal data that is used, concentrates on textual as well as visual data, the latter in the form of images. The current age of digitally afforded multimodality has changed what counts as text and what constitutes reading and writing, as it is easily possible to integrate words with images, sound, music etc (Hull and Nelson, 2005). This multimodal composing is not simply additive but in the best case increases the meaning experienced by multimodal media compared to what each modality contributes (Hull and Nelson, 2005). According to Lahat et al., this is a key property of multimodality called *complementarity*: “Each modality brings to the whole some type of added value that cannot be deduced or obtained from any of the other modalities in the setup” (Lahat et al., 2015).

Multimodal data processing is a challenging task, though. The number, type and scope of potential research questions can be very large due to the diversity in modalities in the data. Additionally, it is a challenge to process heterogeneous datasets in a way that the advantages of each modality are emphasized and the drawbacks minimized (Lahat et al., 2015). To be able to use multimodal data in computations e.g. as input for neural networks, the multiple modalities need to be combined. Sometimes, this is done by simply concatenating the data, other approaches use different ways of fusing the modalities. Sahu and Vechtomova (2019) for example develop a method for automatically fusing these modalities. This method is described in more detail later in this work.

2.4. Technical Background

This section introduces technical concepts and notions used throughout this work. First, the topic of neural networks is broached and afterwards, neural word embeddings are explained, which this work makes extensive use of.

2.4.1. Neural Networks

A feedforward neural network, or Multilayer Perceptron (MLP), defines a mapping $y = f(x, \theta)$ by learning the parameters θ to find the best possible approximation of y for input vectors \vec{x} . As the name suggests, an MLP $f(\vec{x})$ consists of multiple layers which form a chain of functions. Each function in this chain typically looks like

$$g(\vec{x}) = \sigma(W\vec{x} + \vec{b}) \quad (2.1)$$

with a matrix of learnable weights W , a so-called activation function σ , and an optional bias vector \vec{b} . A feedforward neural network typically consists of at least three layers. An *input layer*, a *hidden layer*, and an *output layer*. An example of such a network could look like this:

$$\begin{aligned} f(x) &= g(h(i(\vec{x}))) \\ &= \sigma_{out}\left(W\sigma_h(U\sigma_i(V\vec{x} + \vec{b}_3) + \vec{b}_2) + \vec{b}_1\right) \end{aligned} \quad (2.2)$$

In this example, $i(\vec{x})$ is the *input layer*, $h(\dots)$ the hidden layer(s), and $g(\dots)$ is the *output layer*. If a non-linear activation function σ is used, a multilayer perceptron can solve non-linear problems. Commonly used activation functions include the *Rectified linear unit (ReLU)* function (Nair and Hinton, 2010), *tanh*, or *sigmoid* functions. Given a training objective, the parameters of the network can be learned through backpropagation. A differentiable loss function measures the prediction error which is then backpropagated through the network. The parameters are updated iteratively using a gradient descent method with the goal to minimize the loss for the objective (Bengio et al., 2017).

Over the years, neural networks have seen major improvement and evolution. The MLP holds up well on individual data points but becomes less effective on sequential data like timeseries or natural language. One example of such a more complicated network structure is the Convolutional Neural Network (CNN) (LeCun et al., 1995). This type of networks is very effective in working with images.

2.4.2. Neural Word Embeddings

Neural word embeddings are a way to compute continuous vector representations of words from text corpora. Generally to be able to use words e.g. as input for neural networks, there needs to be a mapping from words to a latent vector space. The naive idea here is to encode words from a corpus as unique one-hot vectors. This, however, has two major drawbacks. First, the larger the number of tokens in the corpus is, the larger will the

vector size be and the more computational effort will be required. Second, these vectors do not contain any information apart from which word they represent. Therefore models were developed that are able to represent words as vectors in a latent vector space where each vector also is able to express information about the context of its word. The concept of word embeddings is that words that are more similar in context will be nearer together in the vector space. One of the best known examples for this is the relation of the pairs “king:queen” and “man:woman”. Word embeddings capture the *gender* relations that express that *king* relates to *queen* in the same way as *man* relates to *woman* (Levy and Goldberg, 2014, Mikolov et al., 2013a).

Word2vec

One of the most impactful innovations in this area has been the development of *Word2vec* by Mikolov et al. (2013a). This approach utilizes a shallow neural network to train word embeddings from large text corpora.

Word2vec utilizes two different architectures. “Continuous Bag of Words” (CBOW) and “Skip-Gram”, the latter of which has later been extended to “Skip-Gram with Negative Sampling” (SGNS) (Mikolov et al., 2013b). Both CBOW as well as SGNS are used in related work of this thesis and in this thesis itself.

The general difference between the architectures is that CBOW is aimed at predicting the current word from its context, and Skip-gram is aimed at predicting the surrounding words from the current word (see Figure 2.2 from Mikolov et al. (2013a)).

2. Background

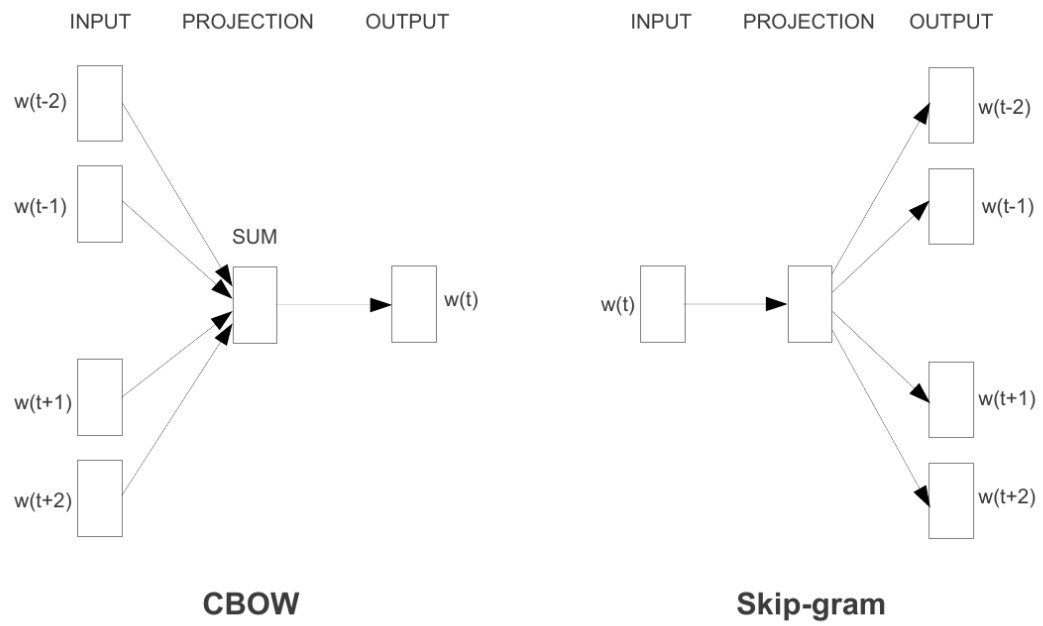


Figure 2.2.: Illustration of the CBOW and Skip-gram model architectures. Image taken from Mikolov et al. (2013a)

3. Related Work

The related work presented here can mostly be categorized into the areas of lexical semantic change detection, multimodal deep learning, and work researching social media texts, especially Twitch.

3.1. Semantic Change Detection

Semantic change detection has been a research topic in linguistics for a long time. More recently with the increasing popularity of natural language processing, automatically detecting semantic change found its way into computational linguistics and computer science.

Survey on Semantic Change Detection in Computational Linguistics

Tahmasebi et al. (2018) conduct an extensive survey on the field of semantic change detection in the natural language processing/text mining community. They investigate the background of this field, approaches to detecting semantic change, as well as evaluation methods, and applications e.g. visualizations.

Their bottom line recommendations are that authors should not simply present numbers as results but rather provide their viewpoint and tell a “story of the word”. Automatically detecting the change in meaning does not always explain how and why the term changed.

Kutuzov et al. (2018) also provide a survey on methods for diachronic word embeddings and semantic shifts.

Diachronic Semantic Change Detection using Neural Embeddings

Recently, most of the work in the field of semantic change detection has been using neural word embeddings, most of the time using Word2vec as initially introduced by Mikolov et al. (2013a). The general approach here is to i) generate word embeddings for different corpora and ii) compute the distance between the vector of a word in one corpus and the vector of that word in another corpus using a distance measure. Most popularly used is the cosine distance to determine the shift between the corpora

Kulkarni et al. (2015) present a rather early work in this field. They compare three methods of increasing complexity. A simple frequency-based approach, a syntactic method using the Jenson-Shannon divergence (Lin, 1991), and a third, distributional method tracking vector representations/embeddings of words over time.

Hamilton et al. (2016b) construct word embeddings using three different methods. Point-wise mutual information (PPMI), singular value decomposition (SVD), and Word2Vec

3. Related Work

using Skip-Gram with negative sampling (SGNS). Their results suggest that SGNS works best for discovering semantic shifts from data but does not perform best for detecting already previously known shifts. The semantic similarity between the vectors of words are measured by the cosine distance. They also introduce two statistical laws of semantic change, i) the law of conformity, stating that frequently used words change at slower rates, and ii) the law of innovation, stating that polysemous words change at faster rates.

In a follow-up paper Hamilton et al. (2016a) introduce a second measure additional to simply calculating the cosine distance of a word’s vectors for two time steps. The second measure is a local neighborhood measure based on the assumption that the change in a word’s nearest semantic neighbors is more relevant than the global shift in a word’s vector semantics. They report that this measure is more sensitive to changes in nouns compared to the cosine measure which is more sensitive to changes in verbs.

Gonen et al. (2020) propose a different neighborhood based metric and are doing novel work by a) using corpora in french and hebrew, and by splitting their Twitter data set by author demographics (Age, Gender, and Occupation) and by day of week (weekday or weekend).

Shoemark et al. (2019) develop a synthetic evaluation framework for semantic change detection systems and apply their framework to 5.5 years of Twitter data. While not exactly comparable to Twitch chat, the fact that such a short time span of text already shows significant semantic change, is an indicator that semantic change happens rapidly on the internet. Their methods of semantic change detection are based on the work of Hamilton et al. (2016a) and Hamilton et al. (2016b).

Synchronic Semantic Change Detection

In contrast to the traditional approach of diachronic semantic change i.e. over time, synchronic semantic change detection examines change not between points in time but between domains in the same time-frame.

Based on their previous work (Kulkarni et al., 2015), Kulkarni et al. (2016) investigate geographic variation of language and word usage in social media, and seek to quantify shift in word meaning across geographic regions e.g. between British and American english.

Ferrari et al. (2017) examine domain-specific ambiguities in meaning on Wikipedia data of different topics using word embeddings and cosine distance.

Schlechtweg et al. (2019) investigate both diachronic semantic change and synchronic semantic change. For evaluation, they use the two gold standards Diachronic Usage Relatedness (DURel) (Schlechtweg et al., 2018) and Synchronic Usage Relatedness (SURel) (Hätty et al., 2019). The diachronic data set consists of two 50 year periods of the german text corpus “Deutsches Textarchiv” (DTA, 2021). The synchronic data set consists of a general web text corpus and a domain specific corpus of cooking-related texts.

In a follow-up paper to Schlechtweg et al. (2019), Kaiser et al. (2021) explore effects of pre- and post-processing on embeddings and find that pre-training helps when the target corpora are small.

3.2. Twitch, Emotes, and Emoji

Twitch has increasingly been used in research in the field of natural language processing. Kobs et al. (2020) find that Twitch.tv chat messages and especially Twitch Emotes carry enough meaning to enable sentiment analysis. Their work sets a precedent for using Twitch chat data in natural language processing and therefore provides relevant previous work for this thesis.

Robertson et al. (2021) apply semantic change detection methods based on Shoemark et al. (2019) to Twitter data with the goal of detecting meaning change in emoji over a timeseries. They provide an interactive website where their results can be explored and downloaded¹.

3.3. Multimodal Deep Learning

Twitch chat consists of multiple modalities. Not only do users see text messages written in the chat window but also the image representations of emotes. A Twitch chat message may therefore consist of both the modalities text and image. In practice, these images can also be moving GIFs which would introduce a video-like third modality. This modality, however, is not explored in this work.

Mouzannar et al. (2018) are using social media posts and multimodal deep learning for damage identification. This can help in crisis situations, natural disasters, or war settings. They are using posts containing images and text and first train images and text CNNs independently. Afterwards, the features of both modalities are simply concatenated and provided as input to a last classifier layer.

Wang et al. (2020) are detecting medical misinformation and anti-vaccine sentiment on social media using multimodal deep learning. While already finished before the outbreak of the Covid-19 pandemic, this work has increased relevancy currently. They use the three modalities text, image, and hashtag as separate modality. Their approach is again to extract features independently and concatenate the representations of the three modalities before a final classification layer. Here, however, they also append a fourth, fused vector to the three modalities in the concatenation step.

Kumar and Garg (2019) are using multimodal twitter data for sentiment analysis. They are considering tweets with raw text, tweets which only contain images, and tweets that contain a combination of text and image e.g. an image with text overlain. They find that text embedded on an image or represented as an image is more expressive for sentiment analysis than both the text-only modality and the image-only modality.

Sahu and Vechtomova (2019) developed a method called Auto-Fusion for multimodal deep learning. This method eliminates having to simply concatenate vectors of multiple modalities and rather lets the fusion network decide which features to concentrate on when fusing modalities. They also introduce a more extensive approach with a fusion network utilizing a generative adversarial network (GAN).

¹<https://emoji-semantic-change.herokuapp.com/> (last accessed 2021-08-17)

4. Methods for Lexical Semantic Change Detection

This section describes the methods used for semantic change detection in this work. Section 4.1 lists tried and tested methods from related work in detail and summarizes how they are used in this work. Section 4.2 introduces a novel, experimental method for semantic change detection by fusing the information of Twitch messages with emotes as a separate modality.

4.1. Semantic Change Detection

The two main approaches to detect semantic change in this thesis are based on the approaches used by Shoemark et al. (2019). Namely **i)** compare the first and the last time step (i.e. months in this case) and **ii)** evaluate an entire time series. The latter method has the advantage that a specific change-point can be detected at which the meaning of a word changes. Their methods, however, also build upon previous work where those measures were introduced and implemented.

4.1.1. Two-step Approach

This is the most common approach to semantic change detection. Word embeddings from two different time steps or domains are taken and compared to each other using a measure. Shoemark et al. use an adapted implementation by Hamilton et al. (2016a,b). Two measures are used in this work to quantify a word’s change between time steps: a) the cosine distance and b) a measure taking into account the neighborhood i.e. the k most similar words in the embedding.

Alignment

To ensure comparability across time, embeddings are being aligned after training using *orthogonal procrustes* as used by Hamilton et al. (2016b). This post-hoc alignment is necessary due to the stochastic nature of Word2vec embeddings. Hamilton et al. (2016b) note that “these methods may result in arbitrary orthogonal transformations” which would preclude comparability of a word across time steps. Hamilton et al. (2016b) define $W^{(t)} \in \mathbb{R}^{d \times |V|}$ as matrix of word embeddings at year t . The alignment across time-steps is then done by optimizing

4. Methods for Lexical Semantic Change Detection

$$R^{(t)} = \arg \min Q^T Q = I \|QW^{(t)} - W^{(t+1)}\|_F \quad (4.1)$$

where I is the identity matrix, $\|\cdot\|_F$ denotes the Frobenius norm, and $R^{(t)} \in \mathbb{R}^{d \times d}$.

Cosine Distance

A straight-forward approach to comparing vectors of word embeddings is by calculating the cosine similarity or respectively the cosine distance. If in the two compared time steps the cosine distance of a word's two vectors is large, it indicates that the same word is located at a different place in the latent vector space. This in turn would indicate that the context of the word has changed and hint at the presence of semantic change.

According to Hamilton et al. (2016a) using the cosine distance as measure for semantic change assigns higher rates of semantic change to verbs.

Neighborhood Measure

Another approach is to construct a second-order vector using the neighbors of a word and comparing the two constructed vectors to see whether the neighborhood of a word has changed (Hamilton et al., 2016a). First, for a time step t and each word in t 's embedding the set of k nearest neighbors is determined using the cosine similarity. For two time steps these two sets are then joined into union set S . The second-order vector v_t for time step t contains entries v_t^i . These entries are created by calculating the cosine similarity of word w and neighbor-word S^i at time t . Finally, the cosine distance is used to measure the distance between the two second-order vectors (Shoemark et al., 2019).

According to Hamilton et al. (2016a) this measure detects higher rates of semantic change in nouns.

Summary Two-step Method

In summary, the two-step approach measures the distance between a word in the embedding of one time step A to the word's embedding in time step B . This distance is calculated either by the cosine distance or the neighborhood measure. The resulting list of words is then ranked in descending order by the distance score. The words which are most likely to have undergone semantic change are therefore ranked at the top.

- **[two-step]** The two-step approach comparing word embeddings from only two points in the timeseries
- **[cos]** Cosine Distance
- **[neighborhood]** Neighborhood Distance Measure

4.1.2. Change-point detection

If a dataset over a time period with constant intervals between time steps is available, it is interesting to monitor the development of semantic change over the entire time series compared to only comparing two points in time. Same as the approach used by Shoemark et al. (2019), the approach used here is based on the method implemented by Kulkarni et al. (2015).

This is done by choosing the word embedding model of one time step t_0 in the time series as comparison model. Possible choices here are “first”, “last” and “previous”. A custom threshold n can be set to include all words that appear in at least n percent of time steps.

Initially, for each word w and each time step t_i , the semantic change score (i.e. cosine distance or neighborhood measure) is being calculated, resulting in a time series of raw distance scores $\mathcal{T}(w)$.

Raw Distances vs. Z-Score

Shoemark et al. (2019) compare the results of using the raw distance values to using normalized z-score values. These z-scores mean how many standard deviations a given word’s distance score is away from the mean of all words’ distance scores in this time step. This method follows Kulkarni et al. (2015) who used it to help control for corpus artefacts. They calculate the mean $\mu_i = \frac{1}{|V|} \sum_{w \in V} \mathcal{T}_i(w)$ and the variance $Var_i = \frac{1}{|V|} (\mathcal{T}_i(w) - \mu_i)^2$ across all words. Then, they transform $\mathcal{T}(w)$ into a *Z-Score* series \mathcal{Z}_i with:

$$\mathcal{Z}_i(w) = \frac{\mathcal{T}_i(w) - \mu_i}{\sqrt{Var_i}} \quad (4.2)$$

$\mathcal{Z}_i(w)$ here is the z-score of the time series for the word w at time step i (Kulkarni et al., 2015). In this work if not otherwise specified, for any change-point approaches and results, the z-score is used as the default.

Comparison Reference Model (First vs. Last)

As mentioned above, the distances of a word in a certain time step are calculated relative to a comparison model. In theory, a third model can be chosen as alignment reference model. Against this third model, both the current model as well as the comparison reference model would be aligned using the orthogonal procrustes method described in Section 4.1.1.

In practice, for this work the two chosen settings are **aligning and comparing** to the first model in the timeseries and **aligning and comparing** to the last model. The configuration where a model is being compared to its previous timestep was discarded as well.

Computing Change-point Results

The change-point approach computes a mean-shift score for each word w and each time-step i . This is done by partitioning the time series $\mathcal{T}_i(w)$ (raw) or $\mathcal{Z}_i(w)$ (z-scores) at i . To get the mean-shift, the difference between the means of the scores in the two partitions is calculated. Shoemark et al. (2019) follow Kulkarni et al. (2015) in using Monte Carlo permutation tests to “estimate the statistical significance of mean-shift scores”. The time-step with the lowest p-value is then chosen as the change-point.



The resulting word list is sorted by mean-shift score (descending) and then by p-value (ascending).

Summary Change-point Approach

In summary, this approach is used to detect semantic change over a series of time steps and calculates a change-point where the meaning or usage of a word is most likely to have changed. The scores are calculated in comparison and alignment to a reference time step which in this work is either the *first* or the *last*. The semantic change scores which are calculated using the methods explained in Section 4.1.1 are either used *raw* or are normalized into *z-score* distances.

- **[cp]** Changepoint
- **[first]** Comparing and aligning all models to the first timestep
- **[last]** Comparing and aligning all models to the last timestep
- **[raw]** Using the raw distance scores
- **[z-score]** Using the normalized z-score distances

4.2. Multimodal Semantic Change Detection

Until now, this work viewed Twitch chat messages exclusively as text. In reality, however, users who participate in or see the chat messages on Twitch.tv see emotes as images. The first thing most people will notice in a message that combines words with emotes is the emote image. This means an emote can not only be viewed as token in a sentence but also as a modifier for the sentence. This is the case with the emote  **Kappa** which can denote sarcasm¹. A sentence like “Nice game!” would therefore have a different meaning if  **Kappa** is appended. This brings up the question whether a semantic change detection system for Twitch should be able to regard emotes as more than just tokens in a sentence. This is what this section seeks to answer.

¹<https://knowyourmeme.com/memes/kappa>

4.2.1. Plan/Architecture

The structure of this architecture is as follows.

First, vocabularies are generated which are explained in more detail in Section 4.2.2. Then Word2vec token word embeddings are generated for a given text corpus which represents a time step or domain corpus. To investigate the influence of Twitch emotes on the semantic change of chat messages, all messages that do not contain an emote are removed.

Emote word embeddings are generated using a separate corpus that spans the entire timespan of the Twitch data set. For the approach using emotes not as tokens but as images, representations for a limited number of emote images are generated using a pretrained convolutional neural network.

Afterwards, using the word-emote-combinations recorded in the vocabularies, the different modalities are fused using the method described in Section 4.2.4. This results in fused vectors representing the vocabulary entries.


Finally, these vectors are then used to calculate semantic change between the time steps using the methods introduced previously in Section 4.1 with minor adjustments to the different format (Gensim Word2Vec models vs. PyTorch tensors).

4.2.2. Vocabularies/Combinations

To find for any token the emotes it occurs with, vocabularies are generated that track the co-occurrence of tokens and emotes. Two separate approaches to generating these vocabularies are explored and for both of them two additional variants. One containing only words (emotes as sentence modifiers), the other containing emotes as well (emotes as tokens and as sentence modifier). In the cases of the vocabularies without emotes, the emotes were also omitted from the word embedding generation.

Global Vocabulary

For each token gather all emotes it occurs with in the given corpus. This creates a mapping of token to a global list of emotes. The two sentences

Nice game  Kappa

Nice  PogChamp

would therefore result in a mapping as seen in Listing 4.1.

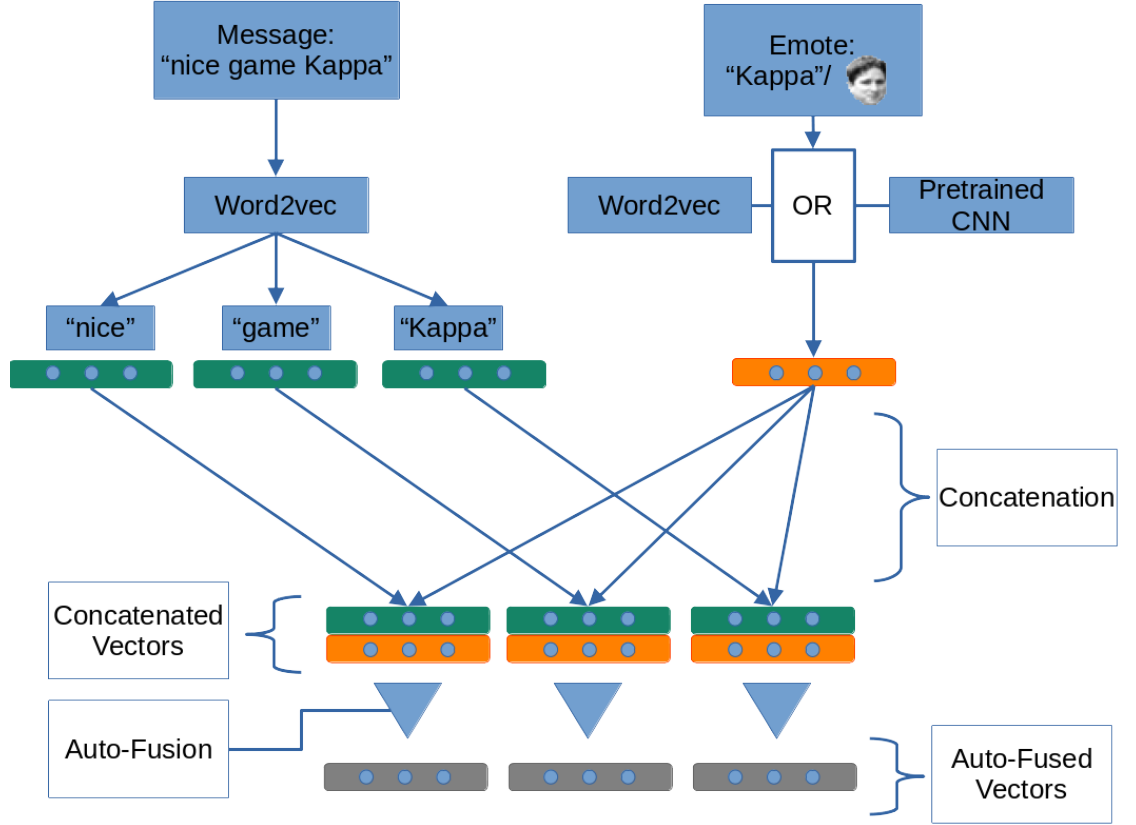


Figure 4.1.: The proposed architecture. A message is seen as two separate parts. The message text and the emote. The latter of which can also be seen as either solely the text representation or the image. The word representations are created via Word2vec. The emote representation is created by either training Word2vec embeddings **only** on emotes or by using a pretrained convolutional neural network. The word and emote representations are then simply concatenated and used as input for the Auto-Fusion network proposed by Sahu and Vechtomova (2019). This network then outputs the auto-fused vectors that will be further used for semantic change detection.


```

"Nice": {
  "emotes": {
    "Kappa" : 1,
    "PogChamp" : 1
  },
  "count": 2
},
"game": {
  "emotes": {
    "Kappa" : 1
  },
  "count": 1
}

```

Listing 4.1: Global Vocab Example. In this example corpus, the word Nice appears twice in total, once with *Kappa* and once with *PogChamp*. The word “game” appears once with the emote *Kappa*

Two potentially separate thresholds n and k filter out tokens that appear less than n times and for each remaining token filter the emote list and remove emotes that appear less than k times with the given token.

Local Vocabulary

This vocabulary gathers any combination of token with a number of emotes resulting in n -tuples of word and co-occurring emotes. If a word occurs with two emotes in one sentence and occurs with one of the two emotes alone in a separate sentence, this will not count towards the same n -tuple.

The three sentences

Nice game 🤔 *Kappa*

Good game 🤔 *Kappa*

Nice 🤔 *PogChamp*

Nice 🤔 *PogChamp* 🤔 *Kappa*

would therefore result in a mapping as seen in Listing 4.2. A threshold k removes any n -tuple that occurs less than k times.

4. Methods for Lexical Semantic Change Detection

(game, Kappa) : 2,
(Good, Kappa) : 1,
(Nice, Kappa) : 1,
(Nice, PogChamp) : 1,
(Nice, PogChamp, Kappa) : 1

Listing 4.2: Local Vocab Example. In this example corpus, the word “game” appears twice with the emote *Kappa*. The word “Nice” for example appears three times in total, once with *Kappa*, once with *PogChamp*, and once with both of the emotes at the same time.

4.2.3. Emote Representations

If emotes should be seen as separate modality from simple tokens in a sentence, vector representations of emotes need to be generated independently from tokens. Two approaches are possible for this endeavor, either representing the emote as token or as image, both of which modify the entire sentence but possibly differently.

Emote Word Representation

Emote word embeddings are generated using a separate corpus created from the entire available Twitch data set. This corpus contains exclusively emote tokens and is grouped into 30 second blocks. The goal of these embeddings is to have emote vector representations that represent how emotes occur with each other independently from any “normal” words. These embeddings use the same latent dimensions as other embeddings trained in this work. Emote embeddings were trained using both CBOW as well as SGNS.

Emote Image Representation

To compute image representations, a pretrained SqueezeNet (Iandola et al., 2016) model was used which is available through the PyTorch hub¹. The classification layer of this model was replaced by a layer reducing the dimensions to latent output dimensions that can be used with the other representations in this work.

4.2.4. Fusion

After having generated the different vocabularies as well as separate representations for each modality, these modalities need to be combined. This combination step utilizes the *Auto-Fusion* architecture introduced by (Sahu and Vechtomova, 2019).

Concatenation Step

The concatenation of the two modalities *token* and *emote(s)* is not entirely trivial. While a word or token is represented by a single word vector, this token can occur with multiple emotes depending on whether the local or global vocabulary is used. To solve this, the

¹https://pytorch.org/hub/pytorch_vision_squeezenet/

emote vectors are averaged. Taking the example of token *Nice* and emotes 🤪 PogChamp , 🤪 Kappa from Listings 4.1 and 4.2, the average of the vector representations of 🤪 PogChamp and 🤪 Kappa is taken and then concatenated with the vector representation of *Nice*.

It might also be that due to the limited number of emote image files or the slightly different threshold when training the emote embeddings, an emote's representation is not available. In this case, the emote is replaced by the token UNK_EM and its vector by a vector of zeros.

Fusing and Training

After creating the concatenated input vectors, these are passed through the Auto-Fusion module shown in Figure 4.2). Tensors $z_{m_i}^{d_i}$ of separate modalities m_i and dimensions d_i are initially concatenated resulting in a vector z_m^k . The dimensions d_i are of same size in this work. This concatenated vector is put through layer \mathcal{T} which produces the initial fused vector z_m^t . This fused vector is then passed through another layer F_c which tries to reconstruct the initial concatenated vector z_m^k from the fused vector z_m^t . Mean squared error is used to calculate the loss of information between the initial concatenated vector and the fused vector.

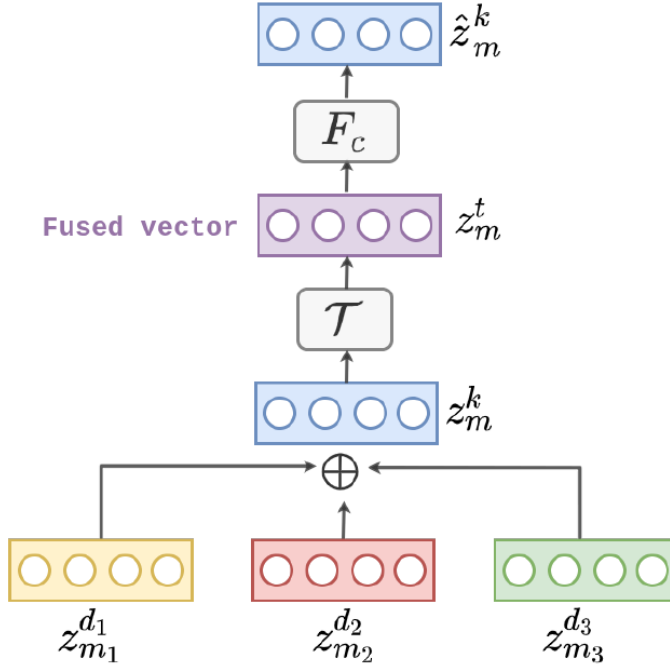


Figure 4.2.: The Auto-Fusion network architecture proposed by Sahu and Vechtomova, 2019. In the case of this work, only two modalities are used instead of three and the dimensions d_i are of same size. This means in this work, two vectors $z_{m_1}^d$ and $z_{m_2}^d$ are fused.

5. Data

This chapter provides information on the datasets used in this work. Apart from the main focus which is semantic change detection on Twitch data, this work uses data from additional sources to compare or verify the methods.

Section 5.1 gives information on the non-Twitch datasets used in this work. Section 5.3 introduces general information about the Twitch dataset. Section 5.4 describes which preprocessing steps have been taken on the Twitch dataset and which subsets of this dataset have been used. Finally, Section 5.2 describes how the synthetic evaluation framework introduced by Shoemark et al. (2019) is being applied to the data in this thesis.

5.1. Datasets for Reproduction Experiments

To validate and test the approaches either introduced in this thesis or by related work, some additional data has been used. This primarily includes several configurations of corpora from an archive of german texts.

5.1.1. Deutsches Textarchiv (DTA)

The Deutsches Textarchiv² (DTA, 2021) dataset is a corpus of german texts ranging several centuries from the 15th to the late 20th century. The dataset contains novels, scientific literature, news articles, and other texts.

For the purpose of semantic change detection, the dataset was split into time steps ranging 50 years. The early time periods from 1450 to 1600 as well as the last time step from 1950 and later contained very little text in comparison to the time steps in between and were therefore discarded.

For this work, the version of 27. July 2020 was used, downloadable from the website of the DTA. The data comes in an XML-like format. Following the preprocessing done for the DUREl dataset introduced in the next Section 5.1.2, only the lemmatized tokens were extracted to construct corpus files containing one sentence of lemmatized tokens per line.

5.1.2. DUREl and SUREl

Diachronic Usage Relatedness (DUREl) (Schlechtweg et al., 2018) and Synchronic Usage Relatedness (SUREl) (Hätty et al., 2019) are two corpora and gold standards for the evaluation of semantic change detection systems.

²<https://www.deutschestextarchiv.de/download#korpora> (last accessed 2021-08-17)

5. Data

DURel uses a subset of the DTA corpus with two timesteps both spanning 50 years, from 1750-1799 (DTA18) and 1850-1899 (DTA19) respectively. SURel uses a corpus for synchronic semantic change detection between a general web text corpus and a domain specific corpus of cooking-related texts.

Both DURel and SURel come with a gold standard of 22 german lexemes for which annotators have determined semantic change.

5.2. Synthetic Dataset

Shoemark et al. (2019) implemented an evaluation framework¹ for semantic change detection systems based on a synthetically created dataset. This dataset contains pseudowords with carefully manipulated frequencies representing artificially created semantic change. The injected pseudowords follow the semantic change schemata introduced in Section 2.2.2. Shoemark et al. (2019) designed seven different schemas, C1 to C3 and D1 to D4. C1-C3 represent actual semantic change which should be detected and D1-D4 represent cases which should not be classified as meaning change by a semantic change detection system.

- C1 This schema’s pseudowords represent the concept of “innovative meaning change” Koch (2016, p. 24) (see Section 2.2.2). A word with a stable sense acquires a new sense.
- C2 Over time these pseudowords change their meaning from an initial sense that vanishes towards a new sense. This represents the cycle of genesis and disappearance as described in Section 2.2.2.
- C3 In this schema, a word with multiple senses drifts towards a single sense. This is not yet a full reductive meaning change (see Section 2.2.2) as the non-dominant meanings have not disappeared yet.
- D1 A word becoming more frequent but not changing in its meaning, i.e. it does not change its co-occurrence.
- D2 A word with two senses, one of which spikes in frequency at one point, e.g. in the context of social media due to a viral event.
- D3 Similar to schema D2 this has two senses but with periodic frequency spikes instead of a single spike, for example the word “turkey” spiking in the USA on Thanksgiving. At any other time, “turkey” means the country of Turkey.
- D4 These pseudowords can have multiple senses, similar to C3, but none of the senses gets more prominent over time.

¹<https://github.com/alan-turing-institute/room2g1o>

5.2.1. Synthetic Dataset Generation Framework

Shoemark et al. (2019) generate their synthetic dataset from their Twitter data using the following steps:

- Sample 10% of one month in the middle of their 5.5 year of Twitter data
- Get word frequencies of this dataset
- Use the Python framework NLTK’s¹ implementation of WordNet (Fellbaum, 1998) to get senses, hypernyms, and hyponyms of words
- Design pseudowords using a chosen number of time steps (e.g. months) and the above information. These pseudowords are created from real words and will replace their “counterpart” in the synthetic dataset.
- Create the dataset by sampling 70% from the original 10% sample for as many timesteps as needed. In these timesteps replace the original words with their pseudoword representation

In total, this approach inserts 210 pseudowords, 90 for C1-C3 and 120 for D1-D4. For further detail on the implementation of the synthetic dataset creation refer to Shoemark et al. (2019).

5.2.2. Synthetic Datasets in this Work

For this work, multiple variants of synthetic datasets were generated depending on the needed usage.

The main synthetic dataset used later in this work in Section 6.4 represents 12 months of Twitch data with synthetic semantic change. The chosen month was November 2019 which is roughly in the middle of the Twitch dataset from May 2019 to April 2020, 10% of which resulted in 130 million words and 5 million unique tokens. In Section 6.4.3 a synthetic dataset generated from DTA data is used. This dataset is lemmatized and therefore contains around 37 million tokens overall but only 38,640 unique ones. Since NLTK’s WordNet does not contain German language words, **Open-de-WordNet** (Siegel, 2021) was used to gather senses, hypernyms, and hyponyms.

For a quick experiment on whether the synthetic dataset generation works similarly well for a synthetic 1 year period versus a longer period, a synthetic Twitch dataset spanning 3 years was generated.

Additionally, to create a baseline for the multimodal approach, another synthetic Twitch dataset was created from data that was preprocessed in the way mentioned in Section 5.4.5.

¹<https://www.nltk.org/>

5.3. Twitch Data Analysis

As this work builds upon the previous related work of Kobs et al. (2020), the data used here has been acquired in the same way by running a crawler on Twitch. The Twitch dataset spans exactly one year from May 2019 to April 2020 (both inclusively). This year spans the beginning of the Covid-19 pandemic, a global event that is expected to leave its mark on the data.

Of each of the months 23-31 days of data is available with generally 288 files per month as the crawler writes files in 5 minute intervals ($288 \cdot 5min = 24h$). Depending on the time of day, these files contain a varying number of messages due to the “prime time” which starts roughly in the evening in European time until late night in USA’s time zones. During this prime time, more messages are written

Overall, the data set contains data from around 550,000 channels and 30 million unique users which amounts to around 560 Gigabyte of data. In plain text form containing only message texts, this is still 165 Gigabyte. On average, messages have a length of 5.4 tokens but a median length of only 3 tokens. This is due to the large number of very short messages, often only containing a single token such as an emote. In total, the data set contains 5.5 Billion messages and 1.6 Billion tokens which are (correctly written) emotes.

Table 5.1 shows numbers for each of the 12 months in the data set. Some months, especially 2019-10 are rather sparse in data compared to e.g. 2020-01. In some places, this was accounted for e.g. by implementing a dynamic threshold (instead of filtering all words with word count < 100 , filter out the least frequent 10%). Overall, however, this had no influence on the nature of the tasks at hand.

5.4. Twitch Preprocessing

Each datapoint in the original data set contains a wide range of information, much of which was of no use in this thesis and was therefore discarded.

Twitch messages were not lower-cased or in any other way preprocessed. This was done to preserve the distinctness of emotes and other proper nouns. For example emotes on Twitch are only displayed as images if the text representation is written correctly. Writing e.g. “kappa” would not result in a displayed image. To not convolute these when it comes to the multimodal approach, lowercasing was not applied. Filtering (e.g. of links, user names etc) was done only before calculating semantic change results.

5.4.1. Extracted Information

The raw data which is attached to a single Twitch message is rather extensive. A lot of the fields are not relevant to the research question at hand or do not fit into the scope of this work. To reduce the computational effort, the raw data has been filtered and one comment of the filtered format contains the information shown in Table 5.2. For most cases e.g. word embedding generation, however, this “medium rare” data format has again been reduced to only contain the plain message text.

Month	Recorded days	Msgs (in million)	Msgs/day (in million)	Raw size (in Gb)	Plain text size (in Gb)
2019-05	25	456	18	46	14
2019-06	30	577	19	59	18
2019-07	31	495	16	51	15
2019-08	26	195	7.5	21	5.9
2019-09	25	379	15	38	11
2019-10	23	98	4.2	9.5	2.7
2019-11	28	242	8.7	25	6.9
2019-12	31	619	20	66	19
2020-01	31	692	22	74	21
2020-02	29	545	19	55	16
2020-03	31	627	20	67	19
2020-04	23	534	23	57	16

Table 5.1.: Numbers for each month in the data set. The two months 2019-10 and 2020-01 are the outliers with October 2019 being the month with least and January 2020 the month with the most recorded date.


Column	Example	Explanation
ts	1557223096325	The UTC timestamp of the comment
chid	26538483	The ID of the channel which the message was written in
msg	luonnosta PogChamp FishMoley	The message text including all emote text representations.
emotes	88:10-17	The emote IDs and position of any Twitch emotes in the message. In this case the message contains the emote with ID 88 ( PogChamp) at the position 10-17.
extemotes	bttv566ca00f65dbbdab32ec0544:19-27	The emote IDs and position of any external i.e. FFZ or BTTV emote in the message. In this case the emote FishMoley .
game	RimWorld	The category or game the current Twitch stream is listed under
usid	44947015	The internal ID of the user who wrote the message
sub	False	Whether the user is subscribed to the channel
mod	False	Whether the user is moderator of the channel
emonly	False	Whether the stream chat is in emote-only mode
r9k	False	Whether the stream chat is in “r9k” mode which prevents a user to write a word for word repeated message inside a specified time-frame

Table 5.2.: Information attached to one comment in the dataset

5.4.2. Message Grouping

As mentioned previously Twitch chat messages are often very short, only consisting of a single emote or word. Additionally, at least in large channels, the throughput of messages is very high, sometimes with multiple messages per second. These messages often are not isolated but rather reactions and responses to the current general situation in the chat channel or stream. Messages written at the same time in different Twitch channels obviously have no relation to each other. This indicates the necessity of grouping messages by the channel they were written in as well as a second level of grouping by time.

Three setups were used: **i)** messages are not grouped in any way (ungrouped), **ii)** messages are grouped by channel and each “sentence” represents a distinct block of 30 seconds, and **iii)** messages are grouped by channel and blocks of 60 seconds. The blocks do not overlap. This would be a potential approach for future work. Apart from the grouping, the messages were not preprocessed further.

5.4.3. Emote Corpus

A subset of the Twitch dataset was built by extracting all emotes and discarding any non-emote words. This was done to explore semantic change on emotes without any other tokens, as well as to train the emote embeddings for the multimodal approach. Since many messages only contain single emotes, the emotes were grouped into 30 second blocks which resulted in corpus files of 30 seconds of emotes in a channel per line. This emote corpus consists of 12GB of data.

5.4.4. Games Corpora

To explore domain specific semantic change between games, a sampled subset was generated by filtering messages from specific games. In this work, data from *Dota 2* and *League of Legends* was used due to the similarity between those games. Both games descend from the *Warcraft 3* custom game *Defense of the Ancients*. They are situated in the *MOBA* (Multiplayer Online Battle Arena) genre where two teams of five players compete against each other on a game “board” of similar shape in both games. Despite many differences, both games have several similar game mechanics and therefore are very much suited for this task. This approach is applicable to any two games, though.

Since *Dota 2* is slightly less popular, the amount of messages is also slightly lower in comparison. The league of Legends corpus in this sampled data set counts roughly 1.8 million lines (each line is one 30s block of messages) and 124 million tokens, the *Dota 2* corpus 600,000 lines and 43 million tokens. This was accounted for e.g. by using a dynamic threshold in the two-step approach of filtering out the least frequent 10% of words. This data set is not the entirety of data that is available for each of the games in the full data set.

5.4.5. Dataset for the Multimodal Approach

For the multimodal approach, the Twitch messages were filtered to make sure that only messages are considered which contain at least one emote (either Twitch native or external). The text corpus then consists of files of two columns. One which contains the entire text of a message and a second one that contains only the text representations of the emotes in a message. To have a direct and undiluted mapping of message text to the respective emotes, the messages are not grouped into 30 second blocks.

Emote Images

To build a Twitch dataset containing the two modalities text and image, emotes were downloaded as image files from <https://twitchemotes.com/>, <https://www.frankerfacez.com/emoticons/>, and <https://betterttv.com/emotes/global> using a crawler. In total this amounts to 898 emotes. This number is not exhaustive as it only contains Twitch's global emotes which at the time of this thesis account for 221 emotes. BTTV's global emotes are 57 and the number of FFZ emotes used is 616. This number comes about because the emotes were crawled from the list of most used emotes available via FFZ. As multiple users can upload an emote with the same name, crawling for the top 1000 emotes results in 616 unique emotes.

6. Experiments and Results

This chapter describes the experiments run using the methods described in Chapter 4, explains which parameters and configurations have been used, and presents the resulting outcomes. First, in Section 6.1 the parameters used to train the word embeddings are explained. In Section 6.2 an experiment from Schlechtweg et al. (2019) is reproduced and Section 6.3 explores semantic change on the DTA corpus in a new way and compares the results to the *Jena Semantic Explorer* (JeSemE) (Hellrich et al., 2018, Hellrich and Hahn, 2017). Section 6.4 describes several experiments run using a synthetically generated dataset following Shoemark et al. (2019)’s evaluation framework. This includes both experiments on Twitch data as well as data from the *Deutsches Textarchiv* (DTA). In Section 6.5, previously introduced semantic change detection methods are applied to a selection of words from the Twitch data set for which semantic change is assumed to have happened during the data set’s time frame. Section 6.6 explores diachronic and synchronic semantic change detection in the domain of video games, specifically *Dota 2* and *League of Legends*. Finally, Section 6.7 describes the experiments run for multimodal semantic change detection.

6.1. Word Embeddings

To generate the word embeddings, Gensim’s (Řehůřek and Sojka, 2010) implementation of Word2Vec (Mikolov et al., 2013a) was used. Related Work uses both the *Continuous Bag of Words* (CBOW) approach ((Shoemark et al., 2019), (Kobs et al., 2020)) as well as *Skip-Gram with Negative Sampling* (SGNS) (Hamilton et al., 2016b, Kaiser et al., 2021, Schlechtweg et al., 2019) which is why this work will often compare both approaches.

The parameters which were successfully used in Kobs et al. (2020) were adapted for this work as well which means a vector size of 128 and a window size of 5. The number of “noise words” for negative sampling was 5, in accordance to Kaiser et al. (2021). The number of epochs was chosen with 10. Experiments with running up to 50 epochs show that further training does not radically improve results. Details on this decision are shown in the appendix A.1.

For all other parameters, the defaults provided by Gensim were used unless otherwise specified.

The embeddings for separate time steps (here: months) were trained independently according to (Shoemark et al., 2019) who reported no benefit provided from continuous training (i.e. initializing the embedding of time step t_i using the embedding of timestep t_{i-1}).

6.2. Reproducing Wind of Change

As a preliminary experiment, the two-step approach implemented by Shoemark et al. (2019) was applied to the **DURel** (Schlechtweg et al., 2018) and **SURel** (Hätty et al., 2019) tasks. As introduced previously, DURel is a gold standard for diachronic semantic change detection, and SURel for synchronic semantic change detection respectively. DURel and SURel define a list of expert-annotated words that undergo semantic change respective to the used datasets. The datasets used by (Schlechtweg et al., 2019) are readily available as well as the gold standards¹. The results presented here are compared to results reported by Schlechtweg et al. (2019). The evaluation is done using Spearman’s ρ rank correlation coefficient to compare the ranking of the system to the ranking of the gold standard. This correlation coefficient measures the rank correlation i.e. the statistical dependence in the ranking of two variables. The DURel and SURel gold standard rankings consist of only 22 target words each, which were selected by five annotators as having undergone semantic change.

The settings for these experiments were chosen according to one configuration used by Schlechtweg et al. (2019). Other settings explored by the authors can be seen in Schlechtweg et al. (2019, Table 2).

The entire experiment including word embedding generation, two-step semantic change detection, and evaluation was run five times. These five runs result in the mean, highest, and lowest results reported in Table 6.1 and Table 6.2. It is noticeable that - as should be expected - using the configuration most closely approximating Schlechtweg et al.’s settings, leads to the highest and most competitive results. In general, the results are relatively close to the results reported by Schlechtweg et al., the differences most likely lie in the stochastic nature of word embeddings as well as the small size of the DURel/SURel target word lists with only 22 words each.

This leads to the conclusion that the application of Shoemark et al. (2019)’s implementations is able to reproduce previously reported results.

¹<https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/wocc> (last accessed: 2021-08-10)

DURel	Measure	Spearman m (h,l)
WoC Baseline (SGNS)	Cosine Distance	0.835 (0.872, 0.814)
CBOW	Cosine Distance	0.615 (0.639, 0.604)
	Neighborhood	0.589 (0.607, 0.567)
SGNS	Cosine Distance	0.763 (0.804, 0.739)
	Neighborhood	0.642 (0.658, 0.618)

Table 6.1.: The results of five runs for each of the four settings compared to results reported by Schlechtweg et al.. Vector size=300, epochs=5, window size=5, negative sampling=5, subsampling=0.001, number of neighbors=25. Spearman m(h,l): mean, highest and lowest result.

SURel	Measure	Spearman m (h,l)
WoC Baseline (SGNS)	Cosine Distance	0.834 (0.838, 0.828)
CBOW	Cosine Distance	0.754 (0.774, 0.745)
	Neighborhood	0.727 (0.737, 0.704)
SGNS	Cosine Distance	0.823 (0.855, 0.795)
	Neighborhood	0.5456 (0.583, 0.504)

Table 6.2.: The results of five runs for each of the four settings compared to results reported by Schlechtweg et al.. Vector size=300, epochs=5, window size=2, negative sampling=5, subsampling=0.001, number of neighbors=25. Spearman m(h,l): mean, highest and lowest result.

6.3. Change-point Analysis of German Texts

In another preliminary experiment, this section explores applying the previously defined changepoint detection method to the DTA corpus.

The DTA dataset¹ offers texts from late 15th to mid 20th century. The DUREl gold standard and subsequent papers (e.g. Kaiser et al. (2021), Schlechtweg et al. (2019, 2018)) use two time frames of this corpus from 1750-1799 and 1850-1899.

The Jena Semantic Explorer JeSemE²³ (Hellrich et al., 2018, Hellrich and Hahn, 2017) offers a possibility to look up the semantic change over time of words in different corpora, among them the DTA. The DTA timeframe which is shown there also only ranges from the 1780s to 1900, however, it is broken down into time steps smaller than 50 years.

The experiment described here aims at applying Shoemark et al.’s change-point approach to the DTA corpus and cross-check the resulting changepoint with JeSemE. The timeframe that was used here ranges from 1600-1950 split into seven 50-year-blocks. Earlier and later timeframes contained too few documents to be considered.

The cross-checking and example selection in this experiment was done manually. Potential candidates were determined using the change-point approach with the cosine measure on embeddings generated with CBOW, aligning and comparing to the first time step. Only words in the result list up to rank 100 were considered except for the word “ergeben” which was ranked 184th but has interesting meaning change. Many of the other words which were detected by the change-point approach are either not available in JeSemE (around 50% of words) or have a detected change-point which is not visible in the graphs shown on the site. This might be due to the reduced time frame that JeSemE uses. Overall, there is a large amount of words which has detected change-point 1850 (i.e. time step 1850-1899). Seeing as Schlechtweg et al. use exactly this time step as the second of the DUREl, it is very likely that times of industrialization and other cultural breakthroughs caused many changes also in literature.

The charts shown here are taken from JeSemE and show “[...] the most specific contexts of [word] over time (higher is more specific). Specific contexts are words which appear often with [word], yet not with other words. χ^2 provides a balanced view [...]” (Hellrich et al., 2018, Hellrich and Hahn, 2017).

The year shown in parentheses is the 50 year time step which was detected as change point e.g. “1850” is the time frame from 1850-1899. A small caveat is, however, that the semantic change in some words might not be due to actual diachronic changes in meaning but rather a synchronic change. This could be the case because the DTA consists of documents from different domains over the years.

¹Deutsches Textarchiv. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften, Berlin 2021. URL: <https://www.deutschestextarchiv.de/>

²jeseme.org/

³<https://github.com/JULIELab/JeSemE>

- **Leiter (1850):**

The word “Leiter” can mean several things. The most prominent of which are a) “ladder”, b) “leader” and c) “electrical conductor”. Figure 6.1 shows that with the rising prevalence of electricity, contexts with “Elektrizität” (= electricity) and “Strom” (= electric power) begins to emerge in the 1810s and 1840s respectively. The context with the word “Trommel” is not clear but its disappearance probably attributed to the detected changepoint in the 1850 time step. The context with “Stufe” (= the step of a ladder) decreases but rises again which is to be expected since the “ladder” meaning of “Leiter” never disappeared.

This is a prime example of innovative meaning change/novel word sense.

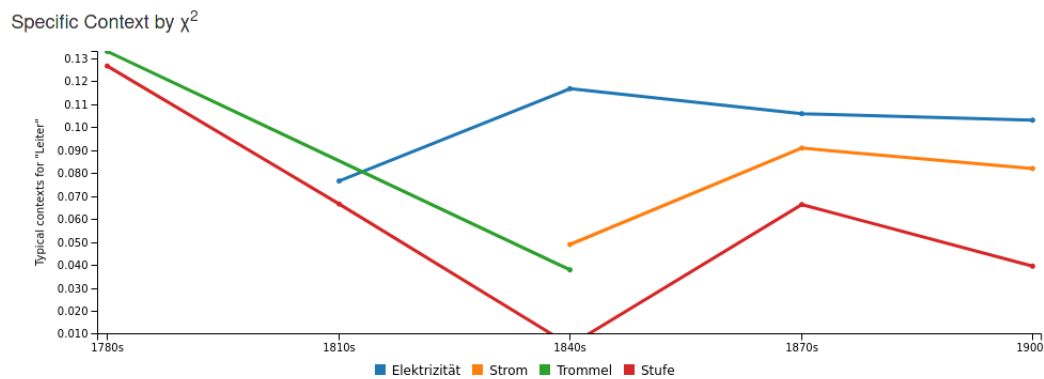


Figure 6.1.: Specific contexts of the word “Leiter” as shown on the JeSemE page. Source: <http://jeseme.org/search?word=Leiter&corpus=dta> (retrieved: 2021-05-16)

- **Einfall (1850):**

The word “Einfall” can mean “sudden idea” as well as “incursion”. This shows in figure 6.2 with the two contexts of “feindlich” (= hostile) and “geistreich” (= ingenious). The latter context rises in prominence from the 1840s on which most likely is the cause of the detected changepoint.

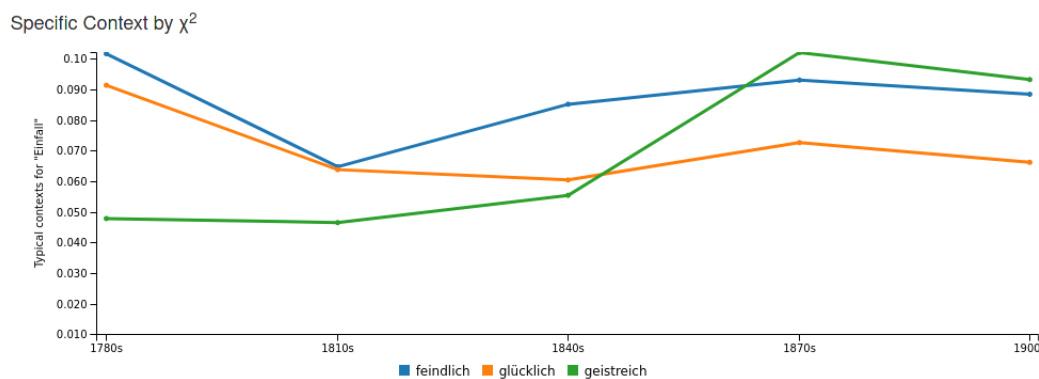


Figure 6.2.: Specific contexts of the word “Einfall” as shown on the JeSemE page. Source: <http://jeseme.org/search?word=Einfall&corpus=dta> (retrieved: 2021-05-16)

6. Experiments and Results

- **ergeben (1750):**

The word “ergeben” can be used as the adjective “loyal” in the context of a loyal servant (= “Diener”). It can also be used as a verb resulting in itself in further different meanings. “Sich ergeben” can be used in the context of “to surrender oneself” or in the context of “something arises as a result” (result = Ergebnis). The latter meaning is the same context as with “hierauf”.

This word is an example of “reductive meaning change” as the context with “Diener” disappears over time.

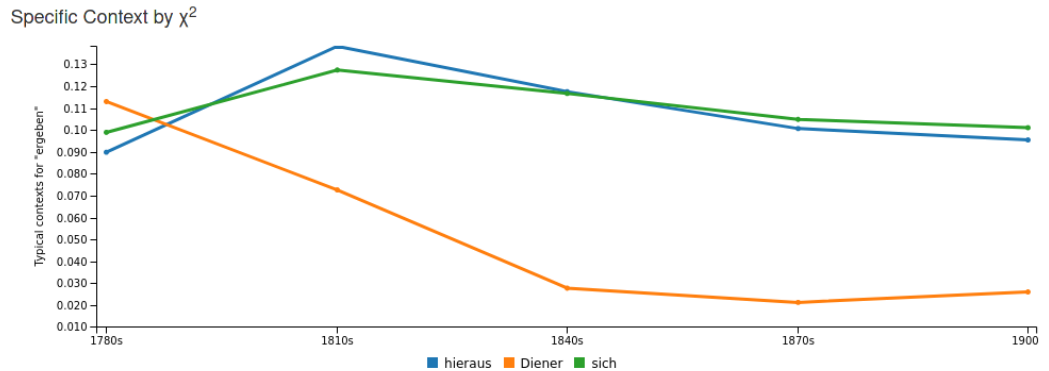


Figure 6.3.: Specific contexts of the word “ergeben” as shown on the JeSemE page. Source: <http://jeseme.org/search?word=ergeben&corpus=dta> (retrieved: 2021-05-16)

- **Schar/Schaar (1800):**

The word “Schar” in general means a “flock” or “crowd” of creatures, be it animals or humans. Figure 6.4 shows a rise in the context with “dicht” (= dense, close together) and a decrease of “schwärmen” (= to swarm). The word “geschlossen” seems to be an old german word for which the exact meaning is not clear.

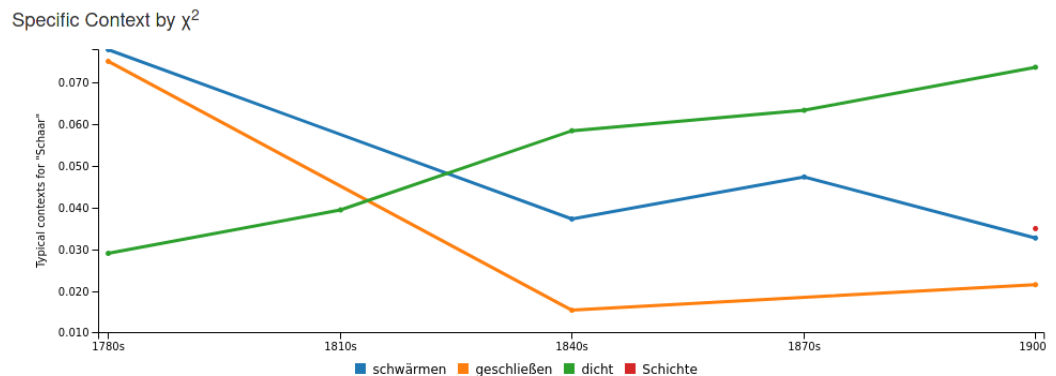


Figure 6.4.: Specific contexts of the word “Schar” as shown on the JeSemE page. Source: <http://jeseme.org/search?word=Schar&corpus=dta> (retrieved: 2021-05-16)

- **Vergleich (1800):**

The word “Vergleich” generally means “comparison”. The changepoint approach seems to agree with JeSemE that a change in semantics occurs in the years after 1800. The usage together with “Preis” (= price) seems to fall out of use and it is increasingly used with the words “im” (= in comparison) and “mit” (= comparison to sth). The context of “stiften” disappears as well, however, it is not clear what the actual meaning here is.

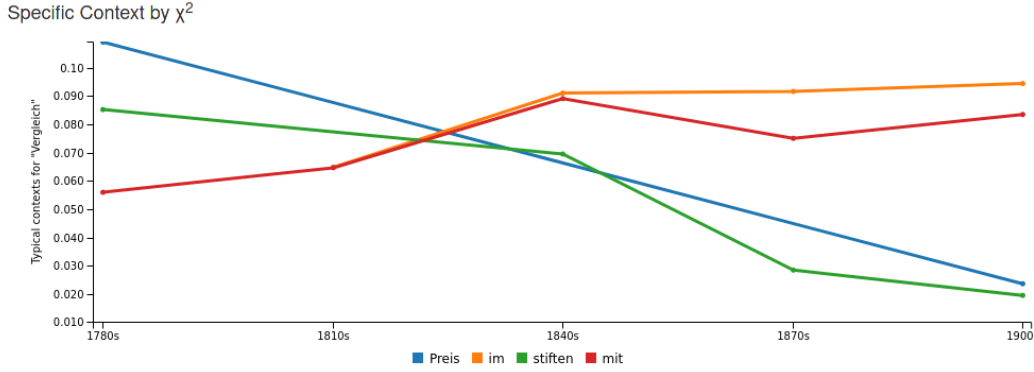


Figure 6.5.: Specific contexts of the word “Vergleich” as shown on the JeSemE page. Source: jeseme.org/search?word=Vergleich&corpus=dta (retrieved: 2021-05-16)

6.4. Synthetic Dataset Experiments

This section explores the synthetic evaluation framework developed by Shoemark et al. (2019). Initially the framework is used to compare preprocessing strategy for Twitch chat messages. Afterwards the results on a synthetic Twitch chat dataset are compared to results reported by Shoemark et al. and a synthetic dataset created from the DTA corpus used by Schlechtweg et al. (2019).

The experiments in this section are mostly aimed at exploring the usability of a Twitch dataset for LSCD. The synthetic data set that was created from Twitch data spans exactly one year, analogous to the 12 months of the real data. Some of the experiments in this section were also tested on a synthetic Twitch dataset spanning 3 years. This was done to check whether such a difference in the synthetic time span has influence on the quality of the evaluation results. The numbers did not show any difference and thus the 12 months were kept.

6.4.1. Evaluation Method

This section describes how the two-step approach (see Section 4.1.1) and the change-point detection approach (see Section 4.1.2) are being evaluated. Given the result lists output by the two methods, Shoemark et al. (2019) calculate the *average precision @K* as their main evaluation measure. They evaluate semantic change detection systems by how highly they rank C1-C3 pseudowords (see Section 5.2).

6. Experiments and Results

The average precision @K (or $AP@K$) approximates the area under a precision-recall curve over the interval 0 to K (Shoemark et al., 2019).

$$AP@K = \sum_{r=1}^K P(r) \Delta R(r) \quad (6.1)$$

The precision $P(r)$ is the percentage of top- r words which are in schemas C1-C3, recall $R(r)$ is the percentage of all C1-C3 pseudowords present in the top- r list. $\Delta R(r)$ is the change in recall between ranks $r - 1$ and r (Shoemark et al., 2019).

Following Shoemark et al. (2019) K is chosen as 50 in the remainder of this work.

6.4.2. Synthetic Twitch Dataset: Comparing Different Message Grouping Setups

In this section, the three chat message grouping setups described in Section 5.4.2 were compared using both the changepoint as well as the two-step approach.

Tables 6.3 and 6.4 show the results of this comparison. It seems that on average for the two-step approach, the ungrouped dataset performs worst and the 30 second grouping performs best. For the change-point approach, however, the ungrouped dataset performs better with the other two grouping variants performing equally less good.

Two-Step		CBOW		SGNS	
		cos	neighbor	cos	neighbor
Twitch	(un-	0.15	0.17	0.14	0.15
	grouped)				
Twitch	(60s)	0.12	0.22	0.18	0.22
Twitch	(30s)	0.14	0.25	0.25	0.27

Table 6.3.: Average Precision @ 50 for two-step approach, comparing the first and last time steps. Noticeable here are the on average lower numbers for the ungrouped approach. This would support the hypothesis that grouping is beneficial to detect meaning and meaning changes.

Change-point	CBOW		SGNS	
	first	last	first	last
Twitch (un-grouped)	0.34	0.40	0.26	0.43
Twitch (60s)	0.25	0.33	0.20	0.32
Twitch (30s)	0.23	0.37	0.22	0.26

Table 6.4.: Average Precision @ 50 for the changepoint approach aligning and comparing to first and last timestep. In contrast to the results of Table 6.3, the ungrouped setting performs best here.

What’s noticeable is the difference in performance between the two-step approach and the changepoint approach when comparing the ungrouped approach to both the 30second and 60second approach. While the ungrouped dataset has on average lower numbers with two-step than the grouped datasets, it shows higher numbers with the changepoint approach.

This warrants a closer look at the detection of pseudowords in the configurations. Tables 6.5 and 6.6 show a comparison of the pseudowords detected using different settings. The two tables have overlapping columns. The purpose of Table 6.5 is comparing the pseudoword detection across the different grouping strategies. Table 6.6, however, compares the differences between semantic change detection methods. The overall number of pseudowords detected does not differ much between two-step and change-point except for the 30s SGNS setting. Split up into pseudoword schemas, however, a different picture unfolds. The pseudoword schemas which show the biggest difference are marked as gray rows.

All approaches succeed for the most part in not detecting pseudowords of schemas D1, D2, and D3. As described in Section 5.2, pseudowords of type D1-D4 should not be detected as they do not represent actual semantic change but rather simple frequency changes or polysemy without a new stable meaning emerging. Apart from that, however, there are substantial differences. Two-step and change-point (last) seem to be rather successful in detecting C2 and C3 pseudowords, with two-step being slightly worse at detecting C2. On the other hand, two-step incorrectly detects a large number of D4 pseudowords as false positives. Change-point (first) does not detect D1-D4 pseudowords but it also is not successful in detecting C3 pseudowords. The grouping of messages (ungrouped vs 30 seconds) as well as the word embedding method (CBOW vs SGNS) does not seem to have a large influence on the overall nature of the results.

6. Experiments and Results

Schema	CBOW				SGNS			
	ungrouped		30s		ungrouped		30s	
	two- step (cos)	change- point (first)	two- step (cos)	change- point (first)	two- step (cos)	change- point (first)	two- step (cos)	change- point (first)
Total	42	35	32	28	43	31	44	26
C1	0	7	0	6	1	7	4	6
C2	11	22	10	17	11	21	11	17
C3	17	3	12	2	16	1	17	0
D1	0	0	0	0	0	0	0	0
D2	0	0	0	0	0	0	0	1
D3	0	3	0	1	0	1	0	1
D4	14	0	10	2	15	1	12	1
AP@50	0.15	0.34	0.14	0.23	0.14	0.26	0.25	0.22

Table 6.5.: Comparison of pseudowords detected in different settings broken down by pseudoword schemas. This table compares message grouping settings.

Schema	CBOW			SGNS		
	two-step (cos)	change- point (first)	change- point (last)	two-step (cos)	change- point (first)	change- point (last)
Total (of 50)	42	35	44	43	31	41
C1	0	7	6	1	7	7
C2	11	22	20	11	21	20
C3	17	3	15	16	1	13
D1	0	0	0	0	0	0
D2	0	0	0	0	0	0
D3	0	3	1	0	1	1
D4	14	0	2	15	1	0
AP@50	0.15	0.34	0.40	0.14	0.26	0.43

Table 6.6.: Comparison of pseudowords detected in different settings broken down by pseudoword schemas. This table contains results from *ungrouped* messages. Noticeable here is the difference in the detection of C3 pseudowords when using the first or the last time-step as reference.

Why do the different settings detect pseudowords differently?

For the definitions of these schemas see Section 5.2. Following are some thoughts on possible reasons why there are differences in the detected pseudowords.

- C1 Except for setting 30s (SGNS) these pseudowords are not detected by the two-step approach. The stable sense might overshadow the new sense so much that the two-step approach does not detect it. Other than that, the reason is not entirely clear.
- C2 Most likely, the two-step approach detects these words because their senses at the start and end point differ a lot. The change-point approach most likely detects these because it actually detects the trend change.
- C3 The two-step approach detects the multiple senses in the starting time step and the new stable sense in the last time step. For the change-point approach, the new emerging sense might not be significant enough to detect it. Shoemark et al. (2019) explain this noticeable difference between comparing to first and last timestep by noting that the representation in the first timestep is very different from the following representations. It is presumed that this makes comparing to the first time step less effective. In the last time step, however, C3 pseudowords have already acquired a dominant and stable new sense which causes the semantic change measures to detect these words.
- D1 Neither the cosine distance nor the neighborhood measure is affected by changes in frequency if the neighboring words stay the same. This is why D1 pseudowords are not detected as candidates for semantic change.
- D2 The two-step approach only takes the start and endpoint into consideration and thus does not notice the spike. The change-point approach most likely detects the spike but due to the implementation of the change-point detection, this is rightfully not detected as semantic change.
- D3 These pseudowords are not detected for similar reasons as with D2.
- D4 The multiple senses fluctuate in frequency from month to month (i.e. it might be different when only taking the first and last month into consideration). This might be the reason why the two-step approach detects these words as semantic change candidates. Over the entire time series this levels out again.

6.4.3. Comparing Synthetic Results: Twitch vs Twitter vs DTA

A secondary dataset was chosen to be compared to the Twitch dataset. This was the **DTA19**¹ corpus used by (Schlechtweg et al., 2019), which contains sentences from the “Deutsches Textarchiv (DTA)”² between the years of 1850 and 1899. Tables 6.7 and 6.8 show a comparison of the evaluation of Shoemark et al.’s synthetic dataset, the synthetic Twitch dataset, and the synthetic DTA dataset. Room2Glo is the title of Shoemark et al.’s paper, which applies their approach to a Twitter data set. It is not entirely comparable because Shoemark et al. (2019) spanned their dataset over a longer period than 12 months. Table 6.7 shows the average precision calculated by the evaluation script when comparing the synthetic “months” 2019-05 and 2020-04.

	CBOW		SGNS	
	cos	neighbor	cos	neighbor
Room2glo/ Twitter	0.25	0.28	-	-
Twitch (un- grouped)	0.15	0.17	0.14	0.15
Twitch (30s)	0.14	0.25	0.25	0.27
DTA	0.23	0.28	0.20	0.24

Table 6.7.: Average Precision @ 50 for the two-step approach.

Table 6.8 shows the average precision @ 50 when using the entire time series. For this approach, only the cosine distance was used, however both aligning/comparing to first and last timestep.

With grouped Twitch messages and the two-step approach, 3 of 4 settings show very competitive results compared to Shoemark et al.’s results on Twitter (see Table 6.7). The same holds true for comparing Twitter and ungrouped Twitch for setting CBOW (first) in Table 6.8

This section showed a general applicability of established semantic change detection methods to the Twitch dataset. It also confirmed the synthetic evaluation framework approach of Shoemark et al. by applying it to the DTA dataset. The overall slightly worse performance of the Twitch dataset can most likely be ascribed to the less structured and more “random” nature of Twitch messages.

¹<https://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/wocc> (last accessed 2021-08-17)

²<https://www.deutschestextarchiv.de/download> (last accessed 2021-08-17)

	CBOW		SGNS	
	first	last	first	last
Room2glo/ Twitter	0.36	0.56	-	-
Twitch (un- grouped)	0.34	0.40	0.26	0.43
Twitch (30s)	0.23	0.37	0.22	0.26
DTA	0.46	0.54	0.39	0.50

Table 6.8.: Average Precision @ 50 for time series approach using changepoint with reference time-step *first* and *last*. All numbers are based on cosine distance results.

6.5. Monitoring Selected Words with presumed LSC

This section takes a look at several words, some of which are related to certain impactful events which happened during the time span of this work’s Twitch data set. These events imply at least an increase in the usage frequency of these words. Here it is examined whether these words also go through semantic change

6.5.1. The Chosen Words and their Frequencies

Kulkarni et al. (2015) use a rather simple frequency based approach as baseline for their semantic change detection system. This captures e.g. sudden spikes in the frequency of a word which might indicate a change in meaning. They calculate the frequencies as seen in equation 6.2.

$$T_t(w) = \log \frac{\#(w \in C_t)}{|C_t|} \quad (6.2)$$

$\#(w \in C_t)$ is the number of occurrences of word w in time step C_t .

This frequency is used for figures 6.6, 6.8 and 6.9 in this section.

The Twitch dataset used in this thesis spans the start of the Covid-19 pandemic (end of 2019/early 2020). This event was discussed worldwide throughout traditional as well as social media and thus it is assumed that Covid had an influence on Twitch chat as well.

The chosen words related to Covid-19 were *lockdown*, *corona*, *quarantine* and *virus*. Some other words such as different spellings (e.g. *Quarantine*, *Corona*) or other related words like *stimulus* went through similar frequency changes but were not considered for this section due to being too infrequent in a fraction of the months.

Figure 6.6 shows nicely that the usage frequencies of the words *corona* and *virus* increase rapidly starting in December 2019 whereas *lockdown* and *quarantine* only start to rise

6.5. Monitoring Selected Words with presumed LSC

from February 2020 on. This frequency graph correlates with the search interest on Google (Figure 6.7). This figure also shows a delayed increase for *lockdown* and *quarantine*.

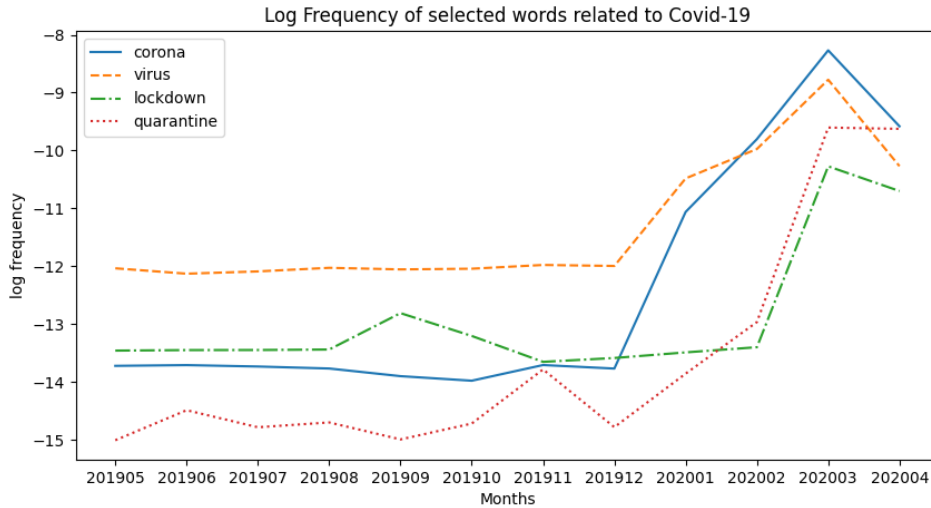


Figure 6.6.: Log frequency of selected Words related to Covid in the Twitch data set spanning 12 months

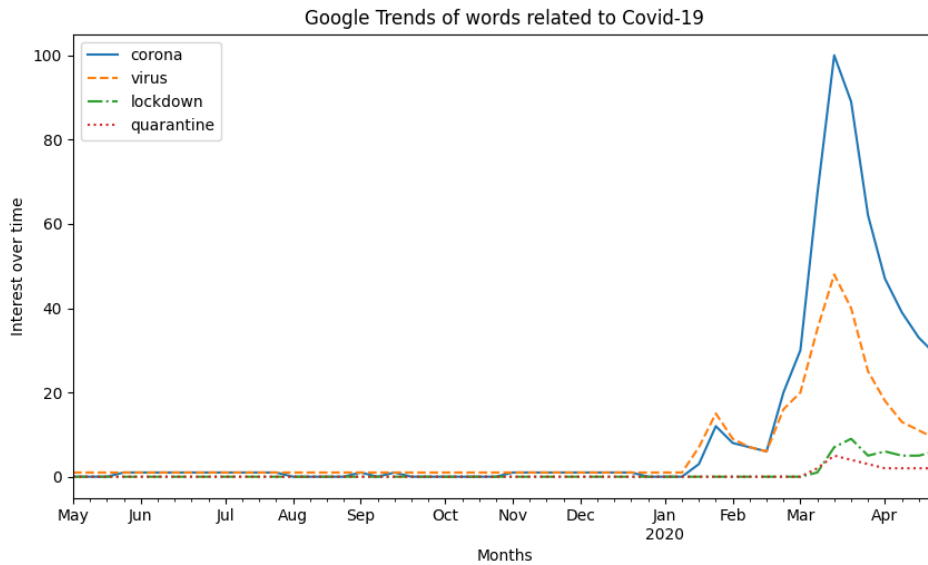


Figure 6.7.: Google trends of selected words related to Covid in the same time period as the Twitch data set. The y-axis represents the interest of time as explained by Google: “Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means that there was not enough data for this term.”

6. Experiments and Results

Another event that resulted in very noticeable frequency changes was the so-called *Blitzchung Controversy*¹. In a tournament for the esport video game *Hearthstone*, Hong Kong pro player *Blitzchung* voiced his support for the protests occurring at the time in Hong Kong. This resulted in punishment for the player by *Hearthstone*’s publishing company *Blizzard Entertainment*. This punishment in turn resulted in intense public response by online communities and media alike, some even calling *Blizzard*’s actions as censorship. Words chosen as related to this event are *hongkong*, *blizzard*, *revolution*, *china*, *hong*, *kong* and *censorship*.

Figure 6.8 shows a large increase in frequency in October 2019 for most of these words.

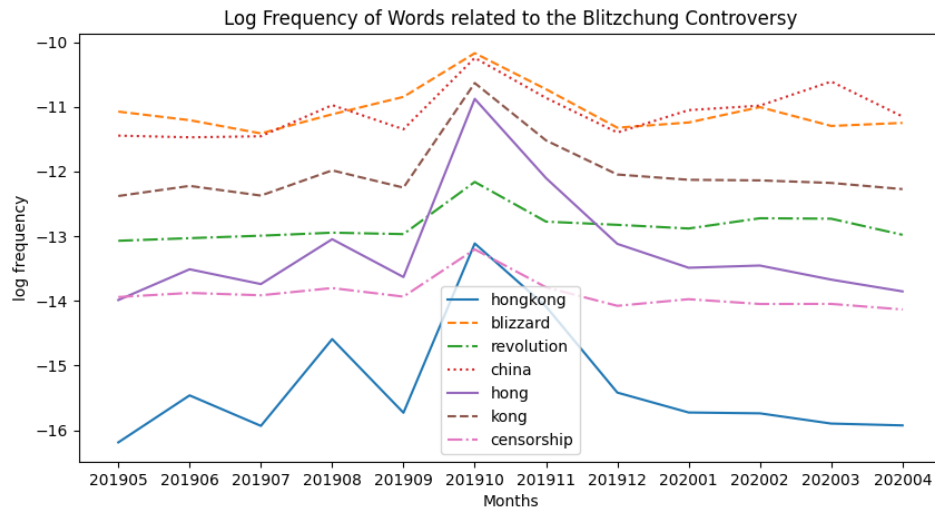


Figure 6.8.: Selected Words related to the Blitzchung Controversy

Some other words were chosen independently of events. These are popular memes or inside jokes on Twitch and related online communities. The word “mald” is an amalgamation of the words “mad” and “bald”. This word rose in popularity when Streamer *Forsen* looked it up on UrbanDictionary² on June 1st, 2019 and spread across Twitch and Reddit from thereon³.

The word “simp” is a controversial term used by some anti-feminist online communities⁴. It is used to make fun of and harass males who perceivedly act submissive to female Twitch streamers. It rose in popularity all throughout 2019 and 2020 and was eventually mentioned by Twitch as a word which may result in a ban when used as harassment⁵.

The word “boomer” is related to the catchphrase *OK Boomer*. This phrase is often used

¹https://en.wikipedia.org/wiki/Blitzchung_controversy (last accessed: 2021-08-12)

²urbandictionary.com

³<https://knowyourmeme.com/memes/maliding> (last accessed 2021-08-18)

⁴<https://knowyourmeme.com/memes/simp> (last accessed 2021-08-18)

⁵<https://twitter.com/twitch/status/1339764206074167296?lang=en> (last accessed 2021-08-18)

as response to statements of older people and as the page KnowYourMeme¹ describes “to mock and debate opinions offered by baby boomers and older people in general”².

Figure 6.9 shows the mentioned “starting point” of *mald* in June 2020 as well as a rising usage of *simp* and a relatively flat peak of *boomer* in November 2019.

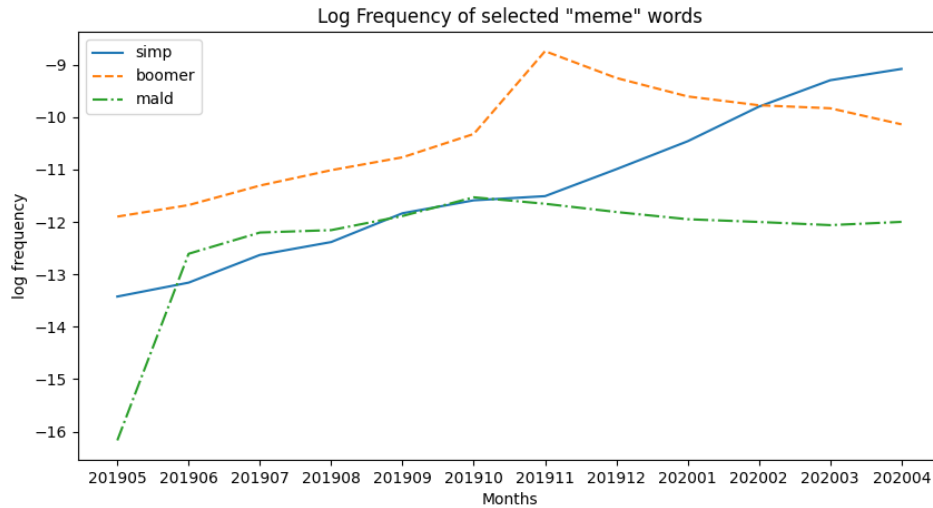


Figure 6.9.: Selected “meme” words

6.5.2. Change-point Detection of Chosen Words

This subsection examines the change points of the words defined in section 6.5.1 detected by the change-point algorithm. Overall, eight different configurations were ran and the resulting change points compared. Ungrouped vs. 30 second, SGNS vs CBOW, and last vs first time-step as reference model. Table 6.9 shows the change points detected by the different settings.

Looking at frequency changes and viral events may yield a hint at where a shift in meaning of words might happen. It does not, however, imply the actual existence of semantic change. This is illustrated by the example of words related to the Blitzchung controversy. The obvious spike in frequency also relates to what Shoemark et al. (2019) defined as their pseudowords of schema D2 (see Section 5.2). This is something a LSCD system should rightfully not detect as semantic change. Similarly, the “meme” words shown in Figure 6.9 are words which could be sorted into Shoemark et al. (2019)’s pseudoword schema D1. They increase in frequency over time but do not change their meaning. An unexpected global development such as an approaching viral pandemic, however, leads to changes in word usage everywhere. These changes do not exclude a social media platform like Twitch.

¹knowyourmeme.com

²<https://knowyourmeme.com/memes/ok-boomer> (last accessed 2021-08-18)

6. Experiments and Results

Generally speaking, the configurations which used the first time-step as reference model found a later month as change point whereas *last* found the earlier month. Table 6.9 shows very well that the system mostly detects December 2019 and January 2020 as change-points for *corona* and *virus*. For *lockdown* and *quarantine*, the system detects mostly February and March 2020. Looking back at Figures 6.6 and 6.7, this is very much in line with the rise in frequency of each of the words. For the other selected words, the system does not detect change-points which are as clear and explainable.

Word	ungrouped				30 seconds			
	CBOW		SGNS		CBOW		SGNS	
	first	last	first	last	first	last	first	last
lockdown	02.2020	02.2020	03.2020	02.2020	-	02.2020	-	02.2020
corona	01.2020	12.2019	01.2020	12.2019	01.2020	12.2019	01.2020	12.2019
quarantine	03.2020	02.2020	03.2020	02.2020	02.2020	02.2020	02.2020	02.2020
virus	-	12.2019	02.2020	12.2019	02.2020	12.2019	01.2020	12.2019
simp	03.2020	05.2019	03.2020	09.2019	03.2020	09.2019	02.2020	-
boomer	-	-	-	-	-	07.2019	12.2019	05.2019
mald	10.2019	05.2019	06.2019	05.2019	10.2019	05.2019	08.2019	05.2019
hongkong	-	05.2019	-	05.2019	-	05.2019	07.2019	05.2019
hong	-	11.2019	-	-	06.2019	-	-	-

Table 6.9.: Detected change-points for selected words.

6.6. Lexical Semantic Change in Games

Different games on Twitch use different language. In single-player first person shooter games, the word “*Sniper*” might be used in the context of a sharpshooter. In a multi-player game such as *Fortnite* it might mean so-called “Stream Snipers” which are players trying to get into the same game lobby as a streamer. In the game *Dota 2*, Sniper is a hero i.e. a playable character in the game. This section explores some combinations of games in an experiment on synchronic semantic change.

The data used here is a sampled dataset where all data from the 12 months is split into games and messages are grouped into 30 second blocks.

6.6.1. Dota 2 vs. League of Legends

Dota 2 and *League of Legends* (or short: *LoL*) are two similar games of a genre often called *Multiplayer Online Battle Arena* or short *MOBA*. This genre originates in the *Warcraft 3* custom game *Defense of the Ancients*. Due to their similarities they have several words already in common which would most likely show semantic differences when compared to non-gaming texts. One example could be “farming” which has a specific context in these games but would be associated with agriculture outside of a gaming context. Another could be the word “creep” which is the term for several non-player characters in both games but has a different meaning outside of these games.

Despite these similarities, both games and their communities use terms which might be used in both *Dota 2* and *LoL* twitch streams but have different meaning. Domain experts for *Dota 2* and *League of Legends* determined a list of example words that succeed in showing semantic differences between both games. Only words up to rank 20 were considered to find the words most likely to actually undergo semantic change and to avoid too much speculation. In total this amounts to 50 non-duplicate words out of 80 (20 out of each combination of CBOW/SGNS and cosine/neighborhood). For context: “champion” and “hero” are the terms for playable characters in *LoL* and *Dota 2* respectively.

Tables 6.10 and 6.11 show two lists of words that were detected by the LSCD approaches. Table 6.10 lists words which have domain-specific context in both of the games but potentially different meaning. Table 6.11 lists words which have domain-specific meaning in one of the games but not the other. Since this evaluation was done by domain experts and not automatically, there is potential of it being subjective. Table A.1 in the Appendix lists words that are more open to discussion than in the previous tables, i.e. words which definitely have a specific context in one game but different context in the other game is not clear.

In total, the domain experts gathered 14 words which have very clear context and usage differences between *LoL* and *Dota 2*. 21 words were found to have specific context but usage or semantic differences were not clear enough. The remaining 15 words were determined as “normal” words such as *already*, *avg* (= *average*), *blank*, *frozen* etc.

6.6.2. Game-specific Time Series - Dota 2

In another experiment, the chat messages for specific games were extracted for each month separately. The change-point approach was then applied to possibly detect game-specific semantic change. The words listed in this section were carefully chosen by the author for manually verifiable semantic change. Thus, the change-point algorithm’s output might contain several words which underwent semantic change that was not verifiable by the author. Table 6.12 shows a small selection of words that are very likely to have actually undergone a shift in meaning and the author’s explanation which is most likely the reason.

6. Experiments and Results

Term	Rank	Explanation
Razor/razor	3	In Dota 2 “Razor” refers to the hero of this name. In LoL this might refer to the item <i>Bloodrazor</i> which was still present in the data set at hand but was since removed in a patch in November 2020.
sticks	5	In LoL this word refers to the champion <i>Fiddlesticks</i> . In Dota “sticks” most likely refers to the item <i>Magic Stick</i>
blitz/Blitz	5	William “Blitz” Lee is a former Dota 2 pro player, coach and esport personality. In LoL this abbreviation refers to the champion <i>Blitzcrank</i> . This is an example of a word that has specific contexts in both games which are unique to the game.
RP	7	In Dota 2 “RP” is used as abbreviation for the hero <i>Magnus’s</i> skill <i>Reverse Polarity</i> . In LoL “RP” refers to an ingame currency formerly called <i>Riot Points</i> which can be used to buy new champions or ingame cosmetic items.
s4	8	<i>s4</i> is a well known Dota 2 Esports pro player. In LoL, this is an abbreviation and refers to either <i>Season 4</i> or <i>Silver IV</i> . Season 4 is the fourth ranked season in the game with one season referring to one year, in this case 2014. Silver IV is a rank in the game that can be reached by players through ranked games.
Phoenix	8	<i>Phoenix</i> is a hero in Dota 2. For the LoL World Championships 2019 a song with the title <i>Phoenix</i> was released. As of May 2021 the music video has 100 million views on YouTube which makes it very likely that this song was referenced often in Twitch chat.
stalker	9	In Dota “stalker” is often used as abbreviation for the hero <i>Nightstalker</i> . <i>Stalker’s Blade</i> was an item in LoL that was in the game during the timeframe of this dataset. Obviously it might also be used as the natural word “stalker”.

Table 6.10.: Seven example words with specific context in both of the games Dota 2 and LoL. **Rank** is the rank in the respective results file the word was found.

Term	Rank	Explanation
grant	1	Similar to <i>Blitz</i> , “grant” is an abbreviation used for Dota 2 personality <i>GranDGranT</i> . The non-domain-specific meaning would be the natural language verb <i>to grant</i> .
jinx	8	<i>Jinx</i> is a playable character in LoL. In Dota 2 this word is most likely used as the natural language word <i>to jinx</i> .
infamous	9	“infamous” is a Peruvian Dota 2 team.
honor	12	In LoL <i>Honor</i> is a system incentivizing positive behavior in the game.
machete	12	<i>Hunter’s machete</i> was an ingame item in League of Legends.
clock	18	In Dota 2, “clock” is used as abbreviation for the hero <i>Clockwerk</i>
haunt	18	“haunt” is a skill used by the Dota 2 hero <i>Spectre</i>

Table 6.11.: Seven example words with specific context in one of the games but not in the other. **Rank** is the rank in the respective results file the word was found.

6. Experiments and Results

Term	Changepoint	Detected by	Possible Explanation
c9/C9	Feb. 2020	last	Well-known esports organization <i>Cloud 9</i> announced a new Dota 2 roster on January 29th, 2020 ^{ab}
w33	May 2019	last	Dota 2 pro player <i>Aliwi “w33” Omar</i> left team <i>Chaos Esports Club</i> on May 9th 2019 ^c and later on June 20th joined <i>Team Liquid</i> ^d
Limp	Sept. 2019	first	Swedish team <i>Alliance</i> announced changes to their roster on 26th September 2019 ^e and announced new players (among them <i>Limp</i>) ^f .
Void/void	Nov. 2019	first	With the <i>Outlanders</i> update ^g in November 2019, a new hero called <i>Void Spirit</i> was introduced to Dota. Previously, “Void” was used in the context of another, older hero called <i>Faceless Void</i> .
Madara	June 2019	CBOW	Dota 2 team <i>Ad Finem</i> announced a new roster on June 27th, 2019 including player <i>Madara</i> ^h
minor	March 2020	first	Due to the Covid pandemic, the <i>OGA DotaPit Minor 2020</i> (a Dota 2 tournament) scheduled for April 2020 was cancelled on March 15th 2020 ⁱ

Table 6.12.: Dota timeseries changepoints. The column “Detected by” names the configuration(s) which detected the changepoint month. Changepoints mentioned here have been detected by at least two of the four configurations (SGNS/CBOW × compare to first/last).

^a<https://www.cloud9.gg/latest/cloud9-comes-off-cooldown-and-rejoins-competitive-dota-2/>. (last accessed: 2021-08-18)

^b<https://twitter.com/Cloud9/status/1222504741692899328> (last accessed: 2021-08-18)

^c<https://twitter.com/w33haa/status/1126560064486150145?s=21> (last accessed: 2021-08-18)

^d<https://twitter.com/chaosec/status/1141658438403002369?s=21> (last accessed: 2021-08-18)

^e<https://www.facebook.com/notes/the-alliance/changes-to-alliance-dota-2-roster/2572068859520465/> (Accessed: 2021-05-28)

^f<https://www.facebook.com/notes/the-alliance/alliance-dota-2-welcomes-new-roster/2587731927954158/> (Accessed: 2021-05-28)

^g<https://www.dota2.com/outlanders> (last accessed: 2021-08-18)

^h<https://twitter.com/adfinemdota2/status/1144256709805977601> (last accessed: 2021-08-18)

ⁱ<https://twitter.com/dota2/status/1239301476465274880?s=21> (last accessed: 2021-08-18)

6.7. Multimodal Semantic Change

This section describes the experiments conducted using the methods in Section 4.2, exploring the potential of multimodal semantic change detection using Twitch chat messages.



6.7.1. Synthetic Dataset Baseline

The baseline results on the synthetic multimodal dataset were computed in the same way as previously shown in Section 6.4. The synthetic messages have not been grouped as explained in Section 5.4.5. Word embeddings were generated with a vector size of 128, window size 5, min-count 100, and 10 epochs using both CBOW as well as SGNS. Due to the extensive number of possible combinations, the evaluation configurations were restricted as seen in Table 6.13. The two-step approach compared the first and last synthetic month using either the cosine distance or the neighborhood measure. The change-point approach uses normalized z-scores, cosine distance and the reference model is either **first** or **last**. With the exception of the combination **SGNS + Change-point**, the results are comparable to those shown in Table 6.4 for the **ungrouped** configuration.

	CBOW		SGNS	
	<i>first</i>	<i>last</i>	<i>first</i>	<i>last</i>
Change-point	0.23	0.38	0.09	0.10

Table 6.13.: Average Precision @ 50 for time series approach using changepoint with align/compare to first. All numbers are based on cosine distance results.

6.7.2. Emotes as Words

This configuration sees emotes still as words but additionally as modifiers for a sentence and the words in it. Referencing again the example of the sentence “*Nice game*  *Kappa*”: If we were to analyze the sentiment of this sentence and the words in it, this would change from the positive “Nice game” to a sarcastic, negative expression due to the sarcasm expressed by  *Kappa* .

6.7.3. Emotes as Images

The idea here is to use a pretrained CNN to compute image representations which can then be fused with the word representation. To this end, the pretrained CNN needs to be modified such that it outputs vectors of the same latent dimensions as the word embedding dimensions. The **classifier** layer of the SqueezeNet¹ model was replaced

¹https://pytorch.org/hub/pytorch_vision_squeezenet/

6. Experiments and Results

by an Adaptive Average Pool layer so that the output of the previous **features** layer (dimensions $512 \times 3 \times 3$) is reduced to $512 \times 1 \times 1$. To then be able to further work with these dimensions, word embeddings were trained again, this time using vector size 512. All other parameters stayed the same.

6.7.4. Vocabulary with Emotes vs Vocabulary without Emotes

As emotes do have a text representation, the question arises whether emotes should be seen solely as tokens in a sentence, as sentence modifiers, or as both. The first option has been dealt with in all previous experiments by simply computing the word embeddings. The latter two options are explored with the multimodal experiments. In the configuration without emotes, emotes are filtered out when training word embeddings and in the vocabularies they only appear as modifiers. This is done to cut out the influence of the emote text representations on the word embeddings. In the configuration with emotes, they are present in both the word embeddings as well as the vocabularies that are used for the fusion step.

6.7.5. Configuration numbers

The number of epochs for the Auto-Fusion step was chosen with 15. Experiments with running up to 50 epochs showed that further training does not radically improve the loss after around 15 epochs. Details on this decision are shown in the appendix A.1.

The min count for the emote word embeddings was chosen with 500 as the full emote corpus consists of around 12gb of data in contrast to the ~ 1.8 gb of data for each synthetic “month”. This emote text corpus contains roughly 140,000 unique emotes. As explained in Section 6.7.3, due to the nature of the image representations with a vector size of 512, the word embeddings used for the **Token + Image** fusion were also computed with size 512.

The threshold for including words, emotes, and word-emote n-tuples in both the global and local vocabs was chosen as 100 in accordance to the min count of the word embeddings.

6.7.6. Results

Table 6.14 shows results detected by the Change-point approach using the fused embeddings. Due to the sheer number of different configurations, the only semantic change detection setting that was used was **Change-point** with the **last** time step as reference model and cosine distance. This was done because this configuration yielded the best results in previous non-multimodal experiments. It is noticeable that the results of trying to detect pseudowords are nowhere near either the baseline results computed specifically for this experiment, nor the results reported previously e.g. in Section 6.4. Chapter 7 will discuss possible reasons for this. Some additional experiments were ran on a non-synthetic Twitch dataset, trying to reproduce results reported in Section 6.5 with selected target words. These experiments were also unsuccessful. Simply applying the change-point approach to a fused, non-synthetic Twitch corpus did not yield any patterns that the

author was able to detect. For example, in the 50 top ranked words when using CBOW compared to SGNS, the intersection of words in both lists was as low as 4.

	CBOW		SGNS	
	Word2vec	Images	Word2vec	Images
Global vocab	8 (0.051, 8)	2 (0.004, 1)	10 (0.006, 6)	7 (0.014, 3)
Global vocab no emotes	10 (0.047, 10)	3 (0.003, 2)	8 (0.011, 5)	5 (0.005, 4)
Local vocab	5 (0.032, 5)	0 (0, 0)	4 (0.022, 4)	1 (0.0001, 1)
Local vocab no emotes	14 (0.127, 14)	0 (0, 0)	5 (0.012, 5)	0 (0, 0)

Table 6.14.: Pseudowords detected by the Changepoint approach in each setting up to rank 50. Format of cells: $n(p, k)$ where n = total number of pseudowords up to rank 50, p = AP@50, k = number of C1-C3 pseudowords up to rank 50.

7. Discussion and Future Work

This thesis set out to do three things, namely a) reproducing results of applying established LSCD methods on previously used datasets, b) applying established methods on a novel dataset, and c) developing a novel method adapted to the uniqueness of this dataset.

One challenge with evaluating the results in this work lies in the novelty of the Twitch data set. The synthetic evaluation framework by Shoemark et al. (2019) was very helpful in this case since there are no gold standards annotated by experts available when it comes to diachronic semantic change on Twitch.

Other experiments such as the reproduction of Schlechtweg et al. (2019) (see Section 6.2) are obviously easier to evaluate by simply comparing numeric results. This also applies when comparing results produced by the synthetic evaluation framework of Shoemark et al. (2019).

The experiments which were run on the DTA corpus in Section 6.3 were evaluated by manually comparing results for several words to the meaning trajectory shown by the JeSemE page. The words and their historic meaning trajectory were researched manually and the results discussed by the author. A similar approach was chosen with the experiments on game specific corpora and synchronic semantic change detection in Section 6.6. Domain experts for both *Dota 2* and *League of Legends* were asked to provide their well-founded insights on the results. Nevertheless, this approach does not pretend to be irrefutable truth.

Overall the experiments in this work again confirm that the established methods do work and showed that the Twitch data set is very much usable for LSCD. This shows especially when analyzing the semantic change-points of words related to Covid-19. As expected, the system detected change-points at the end of 2019 and start of 2020 which coincides with the start of the pandemic. The experiments on synchronic semantic/usage change or rather usage differences in two different game domains showed a similar picture of the usefulness of Twitch data.

Kobs et al. (2020) emphasized the impact and helpfulness of emotes on the task of sentiment analysis on Twitch. For the task of semantic change detection, however, emotes seemed to not have as much impact, at least for the research questions that were asked. The approach of handling emotes as sentence modifiers and computing a separate emote representation did not result in an information gain. Rather it seems as if the results were muddled by the fusing and information was lost. As described in the beginning when introducing the concept of multimodality (see Section 2.3), the idea is that when utilizing multiple modalities, the whole should be greater than the sum of its parts. This already is sometimes rather difficult to achieve and was the initial idea of Sahu and Vechtomova's fusion approach (i.e. not only concatenating the representations but training the fusion step). In the case of this work's approach, however, the whole seemed to be even less

7. Discussion and Future Work

than the sum of its parts.

For the case of the emote image representations, two reasons are very likely: a) Most likely it is necessary to actually train a CNN to be able to represent Twitch emotes and actually contain valuable information, and b) the number of emotes that were available as image files was very small (< 1000). This means a very great number of emotes in the data set would be replaced by an UNK_EM token and a vector of zeros. Not much information gain can come from this even though the emote usage on Twitch is very top heavy, true to Zipf’s law.

More successful was the approach of fusing the w2v token representations with the w2v emote representations. Nevertheless, compared to using only the word embeddings, the results are still lower.

Overall, there are multiple possibilities that explain what happened with the multimodal approach:

- a) Too much information gets lost in the preliminary steps (emote image representation, vocabulary generation, concatenation).
- b) The Auto-Fusion module does not learn how to combine the information.
- c) Not very much additional information is provided by treating emotes as sentence modifiers.
- d) The additional information by treating emotes as sentence modifiers has no impact on semantic change detection.

The bottom line is, that the multimodal approach was an interesting experiment although not successful. It is not clear whether the results of this experiment warrant further work on this idea. Despite the limited success of the multimodal approach, there are several other starting points where the work presented in this thesis could be expanded upon. Several approaches for future work are possible.

Preprocessing The message grouping as introduced in Section 5.4.2 did not use any overlap. This, however, is not a realistic representation of messages on Twitch as the stream chat is continuously running while a channel is online. More realistic would be to have windows of X seconds. This would obviously increase the size of the corpus manifold as many messages will be duplicated.

Utilizing more of the Potential of Twitch Data Apart from the message texts and the emotes (and to a certain degree the channel IDs and timestamps for sorting and grouping) this work did not utilize the full extent of data the Twitch corpus offers. There might be potential in taking channels and/or users into account to explore where semantic change on Twitch starts. Are there users or channels which are spearheading semantic change? Do new emotes and memes start out in certain channels and spread to the entirety of Twitch? One potential example of such an event was mentioned briefly in Section 6.5 with the word “mald” which spread throughout Twitch after streamer *Forsen* googled it only **once** while streaming.

Fusing The approach of using multimodal fusing techniques with chat messages and emotes might provide some future work. Most likely though, a thorough review of the idea behind the method is needed. Nevertheless, Sahu and Vechtomova (2019) not only introduce the Auto-Fusion architecture used in this thesis but also an approach using Generative Adversarial Networks (GANs). A more specific and custom training objective for the fusion step might also be needed so that the fusion actually results in an information gain.

Other LSCD work Semantic change detection on Twitter has been explored in related work. A possibility could be to incorporate Jodel, a german microblogging app which focuses on local communication. This means users will only see posts by other users geographically near them. Using both Jodel and a corpus of german Tweets and explore synchronic usage differences between the communities could be a promising approach for domain specific semantic change detection.

Another idea which was discarded relatively early was to view LSCD as an anomaly detection research question. Possibly using a language model and checking whether at some point during text generation the perplexity of the model might indicate that semantic change is present as the model is “confused” seeing a word at the current position.

8. Conclusion

This thesis took a dive into the research area of semantic change detection, a sub-field of natural language processing. This established area was expanded upon by using a data set that had previously not been used for this task. This data set contains chat messages of the live streaming platform Twitch.tv which have several unique characteristics, first and foremost the existence of *emotes*. In experiments, this thesis was able to reproduce results presented by previous work in the field of lexical semantic change detection. This includes experiments on the DTA, a corpus of german texts. Applying the methods from this thesis to the tasks of DUREl and SUREl showed very close numbers, the difference of which can be explained by the stochastic nature of word embeddings and the small gold standard dataset. These tried and tested methods were then applied to the novel Twitch data set and found to be effective. It was found that words in the Twitch corpus can undergo semantic change and more importantly that this change can be detected despite the nature of the chat messages (very short, seemingly random, high percentage of emotes). Both synchronic and diachronic semantic change detection was successfully explored. The former was shown in an experiment where the impact of the *Covid-19* pandemic was found to have caused semantic change in certain words. The latter was shown when comparing the domains of two video games, *Dota 2* and *League of Legends*, where many words were found that carry different meanings in one of those games compared to the other.

A method to take advantage of the unique nature of Twitch chat messages was designed and implemented. This method treats Twitch messages as multimodal data with emotes as sentence modifiers, a second modality additional to the raw text. Previously used methods for semantic change detection were then applied to this multimodal data. Results indicate that this multimodality either does not actually exist or that emotes do not have impact on the semantic change of words.

Listings

- 4.1. Global Vocab Example. In this example corpus, the word Nice appears twice in total, once with *Kappa* and once with *PogChamp*. The word “game” appears once with the emote *Kappa* 27
- 4.2. Local Vocab Example. In this example corpus, the word “game” appears twice with the emote *Kappa*. The word “Nice” for example appears three times in total, once with *Kappa*, once with *PogChamp*, and once with both of the emotes at the same time. 28

List of Tables

2.1. Table taken from Tahmasebi et al. (2018, p. 35). It shows a more fine-grained segmentation of different change types which are investigated in the literature surveyed in Tahmasebi et al.'s paper. These categories cannot always be separated cleanly.	12
5.1. Numbers for each month in the data set. The two months 2019-10 and 2020-01 are the outliers with October 2019 being the month with least and January 2020 the month with the most recorded date.	35
5.2. Information attached to one comment in the dataset	36
6.1. The results of five runs for each of the four settings compared to results reported by Schlechtweg et al.. Vector size=300, epochs=5, window size=5, negative sampling=5, subsampling=0.001, number of neighbors=25. Spearman m(h,l): mean, highest and lowest result.	41
6.2. The results of five runs for each of the four settings compared to results reported by Schlechtweg et al.. Vector size=300, epochs=5, window size=2, negative sampling=5, subsampling=0.001, number of neighbors=25. Spearman m(h,l): mean, highest and lowest result.	41
6.3. Average Precision @ 50 for two-step approach, comparing the first and last time steps. Noticeable here are the on average lower numbers for the ungrouped approach. This would support the hypothesis that grouping is beneficial to detect meaning and meaning changes.	46
6.4. Average Precision @ 50 for the changepoint approach aligning and comparing to first and last timestep. In contrast to the results of Table 6.3, the ungrouped setting performs best here.	47
6.5. Comparison of pseudowords detected in different settings broken down by pseudoword schemas. This table compares message grouping settings. . . .	48
6.6. Comparison of pseudowords detected in different settings broken down by pseudoword schemas. This table contains results from <i>ungrouped</i> messages. Noticeable here is the difference in the detection of C3 pseudowords when using the first or the last time-step as reference.	49
6.7. Average Precision @ 50 for the two-step approach.	51
6.8. Average Precision @ 50 for time series approach using changepoint with reference time-step <i>first</i> and <i>last</i> . All numbers are based on cosine distance results.	52
6.9. Detected change-points for selected words.	56

List of Tables

6.10. Seven example words with specific context in both of the games Dota 2 and LoL. Rank is the rank in the respective results file the word was found.	58
6.11. Seven example words with specific context in one of the games but not in the other. Rank is the rank in the respective results file the word was found.	59
6.12. Dota timeseries changepoints. The column “Detected by” names the configuration(s) which detected the changepoint month. Changepoints mentioned here have been detected by at least two of the four configurations (SGN-S/CBOW \times compare to first/last).	60
6.13. Average Precision @ 50 for time series approach using changepoint with align/compare to first. All numbers are based on cosine distance results.	61
6.14. Pseudowords detected by the Changepoint approach in each setting up to rank 50. Format of cells: $n(p, k)$ where n = total number of pseudowords up to rank 50, p = AP@50, k = number of C1-C3 pseudowords up to rank 50.	63
A.1. This table collects 21 (together with Table A.2) words that have a specific meaning in at least one of the games but for which the domain experts could not determine a specific reason for why they were detected by the system. Rank is the rank in the respective results file the word was found.	86
A.2. Continuation of Table A.1.	87

List of Figures

1.1.	This figure illustrates the change of meaning which the word <i>gay</i> went through over the last century (Kulkarni et al., 2015).	3
1.2.	Screenshot of a Twitch comment section. The language used in the comments is fairly different from common English. In the bottom right, an emote picker helps with selecting an emote. User names are anonymized due to privacy reasons. This chat is in <i>followers-only</i> mode which means only users who are following the channel can post a message.	4
2.1.	A screenshot of a Twitch channel currently live streaming a League of Legends Esport tournament. The current live stream is visible in the middle of the screen. To the left is a column where - if a user is logged in - the currently streaming, followed channels are listed. Without logging in, channels recommended by Twitch are listed. To the right, the stream chat window is visible.	8
2.2.	Illustration of the CBOW and Skip-gram model architectures. Image taken from Mikolov et al. (2013a)	16
4.1.	The proposed architecture. A message is seen as two separate parts. The message text and the emote. The latter of which can also be seen as either solely the text representation or the image. The word representations are created via Word2vec. The emote representation is created by either training Word2vec embeddings only on emotes or by using a pretrained convolutional neural network. The word and emote representations are then simply concatenated and used as input for the Auto-Fusion network proposed by Sahu and Vechtomova (2019). This network then outputs the auto-fused vectors that will be further used for semantic change detection.	26
4.2.	The Auto-Fusion network architecture proposed by Sahu and Vechtomova, 2019. In the case of this work, only two modalities are used instead of three and the dimensions d_i are of same size. This means in this work, two vectors $z_{m_1}^d$ and $z_{m_2}^d$ are fused.	29
6.1.	Specific contexts of the word “Leiter” as shown on the JeSemE page. Source: http://jeseme.org/search?word=Leiter&corpus=dta (retrieved: 2021-05-16)	43
6.2.	Specific contexts of the word “Einfall” as shown on the JeSemE page. Source: http://jeseme.org/search?word=Einfall&corpus=dta (retrieved: 2021-05-16)	43

List of Figures

6.3. Specific contexts of the word “ergeben” as shown on the JeSemE page. Source: http://jeseme.org/search?word=ergeben&corpus=dta (retrieved: 2021-05-16)	44
6.4. Specific contexts of the word “Schar” as shown on the JeSemE page. Source: http://jeseme.org/search?word=Schar&corpus=dta (retrieved: 2021-05-16)	44
6.5. Specific contexts of the word “Vergleich” as shown on the JeSemE page. Source: jeseme.org/search?word=Vergleich&corpus=dta (retrieved: 2021-05-16)	45
6.6. Log frequency of selected Words related to Covid in the Twitch data set spanning 12 months	53
6.7. Google trends of selected words related to Covid in the same time period as the Twitch data set. The y-axis represents the interest of time as explained by Google: “Numbers represent search interest relative to the highest point on the chart for the given region and time. A value of 100 is the peak popularity for the term. A value of 50 means that the term is half as popular. A score of 0 means that there was not enough data for this term.”	53
6.8. Selected Words related to the Blitzchung Controversy	54
6.9. Selected “meme” words	55
A.1. Epoch comparison for word embedding generation. <i>Hamilton</i> here is the <i>Two-step approach</i>	85
A.2. Epoch comparison for Auto-Fusion.	86

Bibliography

- Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep learning*, volume 1. MIT press Massachusetts, USA:.
- Blank, A. (1997). *Prinzipien des lexikalischen Bedeutungswandels am Beispiel der romanischen Sprachen*. Max Niemeyer Verlag.
- DTA (2021). Deutsches textarchiv. grundlage für ein referenzkorpus der neuhochdeutschen sprache. <https://www.deutschestextarchiv.de/>.
- Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database*. Bradford Books.
- Ferrari, A., Donati, B., and Gnesi, S. (2017). Detecting domain-specific ambiguities: An nlp approach based on wikipedia crawling and word embeddings. In *2017 IEEE 25th International Requirements Engineering Conference Workshops (REW)*, pages 393–399.
- Gonen, H., Jawahar, G., Seddah, D., and Goldberg, Y. (2020). Simple, interpretable and stable method for detecting words with usage change across corpora. In *Proceedings of the 58th annual meeting of the Association for Computational Linguistics*, pages 538–555.
- Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016a). Cultural shift or linguistic drift? comparing two computational measures of semantic change. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing. Conference on Empirical Methods in Natural Language Processing*, volume 2016, page 2116. NIH Public Access.
- Hamilton, W. L., Leskovec, J., and Jurafsky, D. (2016b). Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*.
- Hätty, A., Schlechtweg, D., and Schulte im Walde, S. (2019). SUREl: A gold standard for incorporating meaning shifts into term extraction. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 1–8, Minneapolis, Minnesota. Association for Computational Linguistics.
- Hellrich, J., Buechel, S., and Hahn, U. (2018). Jeseme: A website for exploring diachronic changes in word meaning and emotion. cite arxiv:1807.04148Comment: COLING 2018 System Demonstrations (camera-ready version).
- Hellrich, J. and Hahn, U. (2017). Exploring diachronic lexical semantics with JeSemE. In *Proceedings of ACL 2017, System Demonstrations*, pages 31–36, Vancouver, Canada. Association for Computational Linguistics.

Bibliography

- Hull, G. A. and Nelson, M. E. (2005). Locating the semiotic power of multimodality. *Written communication*, 22(2):224–261.
- Iandola, F. N., Moskewicz, M. W., Ashraf, K., Han, S., Dally, W. J., and Keutzer, K. (2016). Squeezenet: Alexnet-level accuracy with 50x fewer parameters and <1mb model size. *CoRR*, abs/1602.07360.
- 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*, Online. Association for Computational Linguistics.
- Kobs, K., Zehe, A., Bernstetter, A., Chibane, J., Pfister, J., Tritscher, J., and Hotho, A. (2020). Emote-controlled: Obtaining implicit viewer feedback through emote-based sentiment analysis on comments of popular twitch.tv channels. *Trans. Soc. Comput.*, 3(2).
- Koch, P. (2016). *2. Meaning change and semantic shifts*. De Gruyter Mouton.
- Kulkarni, V., Al-Rfou, R., Perozzi, B., and Skiena, S. (2015). Statistically significant detection of linguistic change. In Gangemi, A., Leonardi, S., and Panconesi, A., editors, *WWW*, pages 625–635. ACM.
- Kulkarni, V., Perozzi, B., and Skiena, S. (2016). Freshman or fresher? quantifying the geographic variation of language in online social media. In *Tenth International AAAI Conference on Web and Social Media*.
- Kumar, A. and Garg, G. (2019). Sentiment analysis of multimodal twitter data. *Multimedia Tools and Applications*, 78(17):24103–24119.
- Kutuzov, A., Øvrelid, L., Szymanski, T., and Velldal, E. (2018). Diachronic word embeddings and semantic shifts: a survey. *arXiv preprint arXiv:1806.03537*.
- Lahat, D., Adali, T., and Jutten, C. (2015). Multimodal data fusion: an overview of methods, challenges, and prospects. *Proceedings of the IEEE*, 103(9):1449–1477.
- LeCun, Y., Bengio, Y., et al. (1995). Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995.
- Levy, O. and Goldberg, Y. (2014). Linguistic regularities in sparse and explicit word representations. In *Proceedings of the eighteenth conference on computational natural language learning*, pages 171–180.
- Lichtenberk, F. (1991). Semantic change and heterosemy in grammaticalization. *Language*, 67(3):475–509.
- Lin, J. (1991). Divergence measures based on the shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151.

- Mikolov, T., Chen, K., Corrado, G., and Dean, J. (2013a). Efficient estimation of word representations in vector space. cite arxiv:1301.3781.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013b). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.
- Mouzannar, H., Rizk, Y., and Awad, M. (2018). Damage identification in social media posts using multimodal deep learning. In *ISCRAM*.
- Nair, V. and Hinton, G. E. (2010). Rectified linear units improve restricted boltzmann machines. In *ICML*, pages 807–814.
- Pennington, J., Socher, R., and Manning, C. D. (2014). Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.
- Řehůřek, R. and Sojka, P. (2010). 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. ELRA. <http://is.muni.cz/publication/884893/en>.
- Robertson, A., Liza, F. F., Nguyen, D., McGillivray, B., and Hale, S. A. (2021). Semantic journeys: Quantifying change in emoji meaning from 2012-2018. cite arxiv:2105.00846.
- Sahu, G. and Vechtomova, O. (2019). Adaptive fusion techniques for multimodal data. *arXiv preprint arXiv:1911.03821*.
- Schlechtweg, D., Hättty, 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., Schulte im Walde, S., and Eckmann, S. (2018). 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, Volume 2 (Short Papers)*, pages 169–174, New Orleans, Louisiana. Association for Computational Linguistics.
- 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.
- Siegel, M. (2021). Open german wordnet. <https://github.com/hdaSprachtechnologie/odenet>.

Bibliography

- Tahmasebi, N., Borin, L., and Jatowt, A. (2018). Survey of computational approaches to lexical semantic change. In *Preprint at ArXiv 2018*.
- Wang, Z., Yin, Z., and Argyris, Y. A. (2020). Detecting medical misinformation on social media using multimodal deep learning. *IEEE Journal of Biomedical and Health Informatics*, 25(6):2193–2203.
- Williams, J. M. (1976). Synaesthetic adjectives: A possible law of semantic change. *Language*, pages 461–478.

Glossary

CBOW Continuous Bag of Words. 15, 28, 39, 42, 55, 61

CNN Convolutional Neural Network. 14, 61, 66

DTA Deutsches Textarchiv, a corpus of historical german texts.. 31, 39, 42, 51, 65, 69

DURel Diachronic Usage Relatedness, a gold standard for diachronic semantic change detection. 31, 42, 69

JeSemE The Jena Semantic Explorer, a website visualizing semantic change for words in different corpora.. 39, 42, 45, 65

LSCD Lexical Semantic Change Detection. 10, 45, 55, 57, 65, 67

MLP Multilayer Perceptron. 14

NLP Natural Language Processing. 4, 10

SGNS Skipgram with Negative Sampling. 15, 28, 39, 55, 61

SURel Synchronic Usage Relatedness, a gold standard for diachronic semantic change detection. 31, 69

Word2vec Word2vec, a method to create word embeddings introduced by (Mikolov et al., 2013a). 15, 21, 25

Part II.

Appendix

A. Appendix

A.1. Embedding Parameters

Figures A.1 and A.2 show the results of training word Embeddings on Twitch for 50 epochs as well as training the Auto-Fusion model for 50 epochs to find the optimal number of epochs to use in practice.

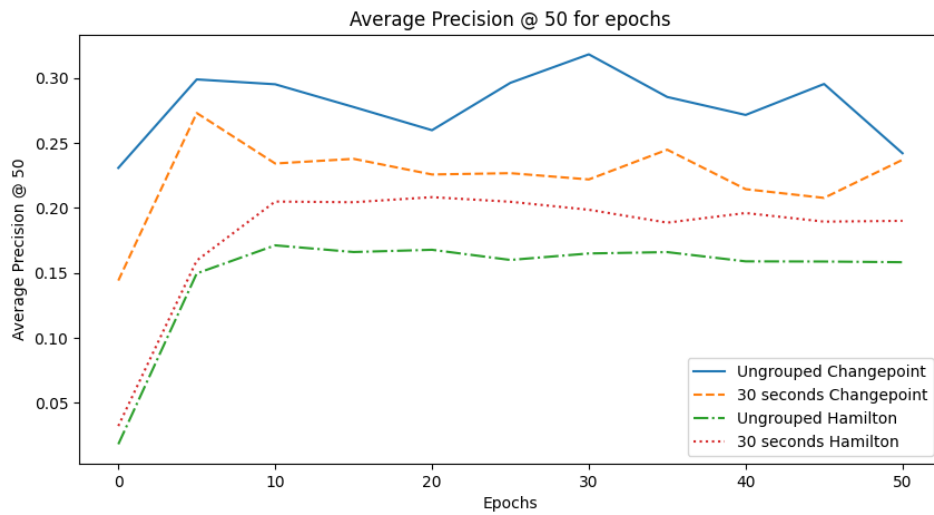


Figure A.1.: Epoch comparison for word embedding generation. *Hamilton* here is the *Two-step approach*

A. Appendix

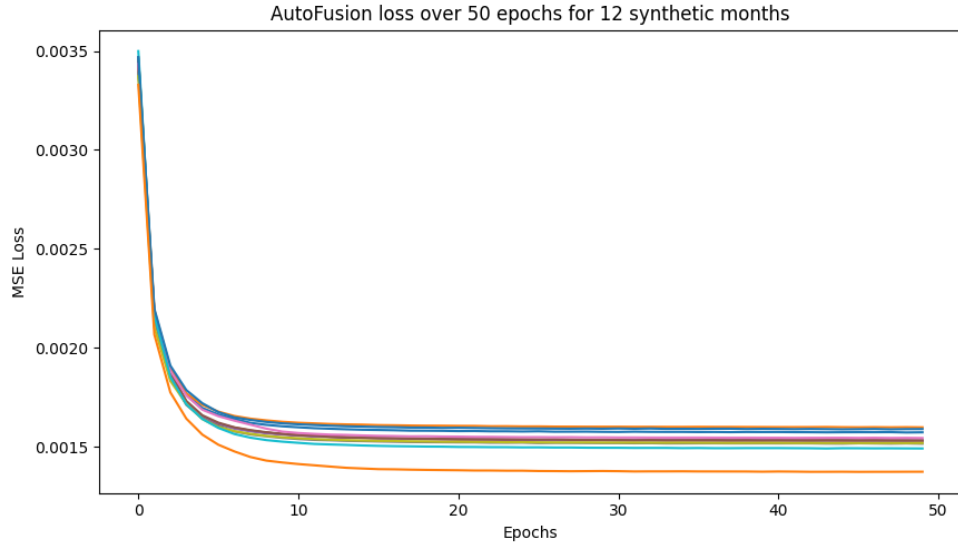


Figure A.2.: Epoch comparison for Auto-Fusion.

A.2. Dota 2 and League of Legends Domain Specific Usage Changes

Term	Rank	Explanation
PETER	4	Peter Dager is a former Dota2 professional player. Any context in LoL is not known
blessing	6	<i>blessing</i> might refer to <i>Aghanim's Blessing</i> in Dota2
DW	10	Might refer to Dota2 hero <i>Dark Willow</i>
lamp	11	Might refer to the item <i>Magic Lamp</i> in Dota2 and possibly to a skill used by LoL champion <i>Thresh</i> .
fishy	15	There are several heroes in Dota2 which are sea-dwelling in their lore.
LP	20	<i>LP</i> in Dota2 refers to the punishment system <i>Low Priority</i> for players with bad ingame conduct.

Table A.1.: This table collects 21 (together with Table A.2) words that have a specific meaning in at least one of the games but for which the domain experts could not determine a specific reason for why they were detected by the system. **Rank** is the rank in the respective results file the word was found.

Term	Rank	Explanation
j4	1	Might refer to Dota 2 professional player Alexei “j4” Lipai. More likely, however, to LoL champion <i>Jarvan IV</i> .
Vladimir	5	Most likely refers to LoL champion <i>Vladimir</i>
leo	6	Might refer to LoL professional player <i>Leo</i> or LoL champion <i>Leona</i>
Chrono	13	Refers to the ingame ability <i>Chronosphere</i> of Dota2 hero <i>Faceless Void</i> .
eve	15	Might refer to LoL champion <i>Evelynn</i> .
Flash	3	<i>Flash</i> is an ingame ability in LoL.
Pick	6	<i>Pick</i> is interesting because it has the same meaning in Dota2 and LoL (picking a hero/champion at the start of a game). Because of the different ingame characters in Dota2 and LoL, the word obviously is used with different context words. This might be the reason why the word appears in this list.
mc	9	<i>mc</i> most likely refers to Dota2 professional player <i>Ivan Borislavov “MinD_ ContRoL” Ivanov</i> whose ingame ID is often abbreviated as <i>mc</i> .
Pinoy	13	Pinoy is a slur used for Phillipine players in Dota2. Dota2 has a large fanbase in South East Asia.
Match	18	This might be a similar case as with <i>Pick</i> previously.
Drums	19	Might refer to the Dota2 ingame item <i>Drum of Endurance</i>
glory	20	Might refer to the LoL ingame item <i>Righteous Glory</i>
imp	1	Might refer to the item <i>Imp Claw</i> in Dota2.
khan	10	Might refer to LoL professional player <i>Kim “Khan” Dong-ha</i>
io	13	Refers to Dota2 hero <i>IO, the Wisp</i>
Random	15	In Dota2 hero selection before a game, a hero character can be picked randomly, i.e. the player will receive a random character to play. This might be a similar case as with <i>Pick</i>

Table A.2.: Continuation of Table A.1.