



DEPARTMENT OF PHILOSOPHY,
LINGUISTICS AND THEORY OF SCIENCE

EXPLORING LEXICAL SEMANTIC CHANGE IN POLISH USING XL-LEXEME

An Analysis of the Parliamentary Corpus (1919-2023)

Ewa Słowińska

Master's Thesis:	30 credits
Programme:	Master's Programme in Language Technology
Level:	Advanced level
Semester and year:	Spring, 2024
Supervisor:	David Alfter, Pierluigi Cassotti
Examiner:	Nina Tahmasebi
Keywords:	LSC, Polish, XL-Lexeme

Abstract

The scope of this thesis is on Lexical Semantic Change (LSC) and its automatic detection in the Polish language. Following [Cassotti et al. \(2023\)](#)'s findings, the following thesis leverages XL-Lexeme, a transformer-based bi-encoder model, to perform LSC detection on the Polish Parliamentary Corpus divided into two time periods: (1) 1919-1961 and (2) 1989-2023. The aim of this thesis is to examine the performance of XL-Lexeme with a Polish dataset and to state what kind of changes occurred between the two predefined time periods. The results suggest a rather robust performance of XL-Lexeme, coinciding with the judgements of a native speaker of Polish, however the influence of context and occasional annotation errors hinder the reliability of the results. The types of changes detected through close-reading include semantic widening and narrowing as well as changes in the meaning distribution, which are often be related to technological and political advancements. Additional WiC task performed on a small portion of annotated sentence pairs further confirms XL-Lexeme's swift handling of Polish language, yielding a precision as high as 0.971 but falling behind on recall which amounts to 0.684.

Contents

1	Introduction	1
1.1	Language Change	1
1.2	Lexical Semantic Change	1
1.2.1	Lexical Semantic Change Detection	2
1.2.2	Applications of LSC Detection	2
1.3	Focus	3
1.4	Motivation	3
1.5	Polish	4
1.6	Research Questions and Hypotheses	4
2	Background	5
2.1	Distributional Hypothesis	5
2.2	Neural Networks	5
2.2.1	FNNs and RNNs	5
2.2.2	Transformers	6
2.3	Automatic Lexical Semantic Change Detection	7
2.4	Embeddings	7
2.5	Literature Review	8
2.6	XL-Lexeme	9
2.7	Algorithms	10
3	Experimental Setup	12
3.1	Dataset	12
3.1.1	Motivation	12
3.1.2	Properties	12
3.1.3	Preprocessing	12
3.2	Extracting Data	13
3.2.1	Lemmas and Word Forms	13
3.2.2	Comparing Lemmas	13
3.2.3	Final Lemma List	14
3.2.4	Random Sentence Selection	14

3.2.5	Sentence Extraction	14
3.3	LSC Detection	15
3.4	Analysis	15
3.4.1	Qualitative Analysis (LSC)	15
3.4.2	WiC Task	15
4	Results and Discussion	17
4.1	Numerical Results	17
4.2	WiC Task	17
4.2.1	Annotation	17
4.2.2	Results of the WiC Task	18
4.3	Qualitative Analysis (LSC)	20
4.4	Noise	21
4.5	Influence of Context	22
4.6	Answers to the Research Questions	22
5	Ethical Considerations	24
6	Critiques and Limitations	25
7	Conclusion	27
8	Future Work	28
	References	29
A	Resources	34
A.1	PRT Top 100	34
A.2	APD Top 100	36

1 Introduction

Language is ever-changing—speakers constantly mould and shape the language system to create new rules that would meet their ongoing needs (Aitchison, 2001). While the oft-cited cultural, societal and technological advancements (Tahmasebi et al., 2021) are doubtlessly essential, the factors steering the language change are far more complex. Beneath the palpable factors, lie equally significant internal motivations, economy and language contact (Mufwene, 2001). Although virtually all language components are susceptible to evolutionary shifts (Nordquist, 2019), the scope of this work is on the so-called lexical semantic change (LSC)—that is, the change in word meaning over a specified period of time (Schlechtweg et al., 2020). The following thesis leverages XL-Lexeme, a transformer-based bi-encoder model (Cassotti et al., 2023), to investigate lexical semantic change in the Polish language from the 1910s to the 2020s.

1.1 Language Change

“Language change” refers to the process of a step-by-step evolution of certain linguistic features (Nordquist, 2019). Although the type and speed of changes involved in this process may be variable in nature—the language of conservative communities, for example, may change more slowly than that used among more progressive communities (Nevalainen et al., 2020)—language change is considered to be a phenomenon universal to all languages of the world. Studying and analysing language change may bring about crucial findings not only in the field of linguistics but also for researchers in history and sociology (De Busser, 2015) or cognitive science (Li et al., 2024). Given that language and culture are virtually inseparable (Rodrigues & Ravasco Nobre, 2020), the study of linguistic change can give researchers irreplaceable insights into the evolutionary shifts that have taken place throughout the history of different societies (De Busser, 2015). Moreover, macro-level language changes often reflect micro-level behavioural preferences, unravelling the mechanisms driving human cognitive processes (Li et al., 2024). Lastly, tracking and analysing language changes fosters efficient communication between generations.

Language changes may affect all language components including sound, lexicon, semantics and syntax (Nordquist, 2019). Some oft-cited examples of language change include the Great Vowel Shift in English (Duignan, 2024), lexical borrowings (Grant, 2015) or gradual pejoration (negative shift) of certain word meanings (Okrent, 2019). There exist many theories uncovering the true motivations behind language change. As researchers claim, social, cultural, political, technological, and demographic factors all play a role in shaping languages (Mantiri, 2010). Colonisation, migration and trade are among the most commonly cited phenomena that directly influence the emergence of language change. A frequent result of the linguistic contact to which these historical phenomena contribute is borrowing or hybridisation at the level of sounds and vocabulary (Mufwene, 2001). Similarly, technological progress adds to changes in language—new findings and inventions introduce the need to invent names for freshly discovered phenomena or machines (Beaumont, 2021). Finally, factors such as social status or fashion, turn out to be equally active in influencing linguistic changes such as the fluctuating trend in the pronunciation of New York’s *r* (Labov, 1997).

1.2 Lexical Semantic Change

The focus of this thesis is on one specific type of language change, namely lexical semantic change. As was mentioned earlier, LSC involves changes in the word’s meaning. One example of lexical semantic change can be presented on the German word *bein*, the meaning of which has shifted from Old High German ‘bone’ to Modern German ‘leg’ (Urban, 2015). As Podlaski (2017) notes, new uses of old word forms can result from, among others, a lack of names for previously unfamiliar things or phenomena (nominative need), as well as the need to express thoughts and emotions more adequately (expressive need). Podlaski (2017) further notes that the first group includes words such as *virus* ‘detrimental piece of code’, while the second

group especially includes those meanings of pre-existing words that qualify as colloquial, e.g. *ass* ‘someone rude’.

The nature of lexical semantic changes can vary, as has been repeatedly studied and confirmed by many well-known linguists such as Bloomfield (1984), Ullmann (1951, 1962) and Blank (1999). Below is a summary of some of the LSC types distinguished by the above-mentioned researchers:

- a) metaphor - change based on the similarity between two concepts (e.g. *mouse* ‘animal’ → ‘device’)
- b) metonymy - change based on relatedness of two concepts (e.g. *crown* ‘object’ → ‘royalty’)
- c) synecdoche - change based on one concept being a part of the other (e.g. *wheels* ‘part of a car’ → ‘car’)
- d) specialisation - change based on a downward shift in a taxonomy (e.g. *corn* ‘grain’ → ‘maize’)
- e) generalisation - change based on an upward shift in a taxonomy (e.g. *jeans* ‘specific brand’ → ‘pants’)

The types of change described above can have a variety of consequences for the semantics of a word. The four types of consequences identified by Ullmann (1951, 1962) are widening, narrowing, amelioration and pejoration of meaning.

1.2.1 Lexical Semantic Change Detection

The lexical semantic change detection task involves extraction of the shifts in the meanings of words over time within a specified language (Tahmasebi et al., 2021). The LSC detection task typically leverages large text corpora divided into specified periods to detect shifts in word usage, connotations, or semantic associations. While, traditionally, lexical semantic change has been studied manually via ‘close reading’, over the past few decades, computational approaches to LSC detection task have begun to rise (Tahmasebi et al., 2021). The main advantage of automatic LSC detection over the manual ‘close-reading’ approach is the ability of computers to process huge amounts of information in a relatively short time. In addition, computer-based LSC detection models are often able to catch more subtle language changes, and their “reasoning” is often more consistent than that found with a human annotator. In other words, automatic LSC detection is much more efficient, robust and often less expensive than manual LSC detection. Naturally, computational approaches to LSC also hold certain disadvantages. One considerable drawback is that developing models for LSC detection requires specialized knowledge in the field on computer science. Moreover, computational approaches do not provide cultural understanding of the processed data and thus may omit certain changes that could be easily noticed by a human annotator. As one can see, both manual and computational approaches can be of use, depending on the specific goals of the research. It is, thus, often advantageous to combine close-reading with automatic LSC detection to leverage their respective strengths.

1.2.2 Applications of LSC Detection

Detecting lexical semantic change serves not only as an end in itself but also as a valuable tool for advancing research across a spectrum of disciplines, including sociology (Wevers, 2019; Garg et al., 2018; Laine & Watson, 2014), cognitive science (Vylomova et al., 2019), and literature (Haider & Eger, 2019; Bizzoni et al., 2020). LSC detection proves to be particularly useful in diachronic studies that aim at recognising and dissecting changes that have occurred throughout predefined periods of time (Schlechtweg et al., 2020; Kutuzov & Pivovarova, 2021). By detecting patterns driving the evolution of thought, LSC not only illuminates the past but also enables us to put forward hypotheses about the future. This subsection provides a brief overview of studies that applied different LSC detection techniques to further develop their research domains.

One noteworthy application of LSC detection techniques can be found within literary studies. Haider & Eger (2019), for example, use an LSC model to explore changing poetic tropes throughout different literary

periods. [Bizzoni et al. \(2020\)](#), on the other hand, focus on scientific writing, revealing numerous changes in the scientific register of the English language. Both of these studies serve as proof that LSC detection is a useful and important tool in humanities, thus calling for a collaboration between text-based research fields and computer science.

LSC detection is often performed to track changes in different societal values. [Wevers \(2019\)](#), for example, uses LSC detection to track the history of gender bias occurring in Dutch new media across the span of forty years. A similar approach is taken by [Garg et al. \(2018\)](#) who utilise LSC detection tools to analyse gender and ethnic stereotypes exhibited by U.S. citizens. Yet another study leveraging LSC detection in the domain of sociolinguistics is [Laine & Watson \(2014\)](#), whose focus lies on linguistic sexism in the popular British magazine, *The Times*.

[Vylomova et al. \(2019\)](#) utilise LSC detection models to study the semantic change of harm-related concepts such as “addiction”, “bullying”, “harassment”, “prejudice”, and “trauma”, thus revealing a general broadening of these terms’ semantic pools. These findings offer psychologists and mental health specialists a new perspective on mental health issues and their history.

1.3 Focus

This thesis investigates the performance of XL-Lexeme in LSC detection in the largely understudied Polish language. Furthermore, the thesis aims to uncover patterns underlying LSC in Poland’s pre-soviet and soviet versus post-soviet eras and to investigate the impact of historical events, socio-political shifts, and technological advancements on the meanings of words. The following thesis constitutes the first empirical study of automatic LSC detection in the Polish language, hopefully serving as a starting point for further investigation into this topic.

1.4 Motivation

LSC detection is of particular interest to fields such as psychology, cultural anthropology, history, literature or philosophy ([Tahmasebi et al., 2021](#)), which base and advance their knowledge through careful inspection of historical texts. Understanding the changes in word meaning is therefore crucial for the scientists of these fields to fully grasp the contents of the documented sources. Through studying and detecting lexical semantic change we are able to analyse and preserve cultural heritage, improve communication and facilitate further research ([Perrone et al., 2021](#)). LSC detection task is also crucial for the entirety of the Natural Language Processing (NLP) field—understanding semantic changes is of particular importance for developing effective language models and search engines, which tend to work with both contemporary and historical data.

So far, the research into LSC detection has been conducted predominantly in the English language ([Schlechtweg et al., 2020](#); [Kim et al., 2014](#); [Mitra et al., 2014](#)). Other languages that tend to reappear in the published works are Swedish, German and Latin ([Schlechtweg et al., 2020](#); [Perrone et al., 2021](#); [Kurtyigit et al., 2021](#)). This clear language bias has several negative consequences including limited generalisability and cross-linguistic comparison of the results, worse model performance in multilingual settings and underrepresentation of linguistic diversity. The choice of the Polish language focus in the following thesis serves as a step toward countering these negative consequences. The choice of the thesis topic is therefore motivated not only by the growing demand for state-of-the-art LSC models but also by the visible lack of linguistic diversity in the field. Similar motivations drive other LSC researchers, including [Periti & Tahmasebi \(2024\)](#) who evaluated models in eight different languages: English, Latin, German, Swedish, Spanish, Russian, Norwegian, and Chinese.

1.5 Polish

Polish is a West Slavic language spoken primarily in Poland, dating back to the 10th century A.C. when the Polish state was established, uniting a few linguistically related tribes ([Britannica, 2024](#)). Throughout the centuries of frequent contact, Polish has been significantly influenced by other languages, especially Latin, French, Italian and German, which gave rise to a high number of borrowings and lexical and semantic shifts ([Knara, 2017](#)). There exists a number of historical events that could have potentially triggered an influx of lexical semantic changes in the Polish language. This thesis zooms in on two time periods in Polish history. First is the interwar period followed by World War II and the establishment of the Soviet-ruled Polish People's Republic and covers the years 1919-1961. The second period starts in 1989 with Poland's first partially free and democratic parliamentary elections since the end of the World War II and ends in 2023. The logic behind the choice of the time periods assumes that the changing forms of government, progressive globalisation as well as advancing technology all contributed to numerous meaningful lexical semantic changes.

1.6 Research Questions and Hypotheses

The following thesis aims to answer two research questions (RQs). Their contents as well as the respective hypotheses (RHs) can be found below.

RQ1: What is the performance of XL-Lexeme in detecting LSC in the Polish language?

RQ2: What kinds of LSC occurred between the pre-soviet and soviet versus post-soviet Polish?

As was previously mentioned, automatic LSC detection has never been performed, or at least documented, for the Polish language. Thus, making hypotheses about the performance of any model proves to be particularly difficult. Any speculations made about the prospective performance of XL-Lexeme must, therefore, originate from examining the results of the original study leveraging this model ([Cassotti et al., 2023](#)). The first study of XL-Lexeme's multilingual performance was done in English, German, Swedish, Latin, and Russian. For each language, except for Latin, XL-Lexeme achieved acclaimed results, proving its multilingual flexibility that could potentially translate to Polish. This hopeful stance regarding XL-Lexeme's performance in the Polish language is further solidified by XL-Lexeme's swift handling of Russian—language, which is closely related to Polish in terms of grammar, vocabulary and history. The hypothesis for RQ1 is therefore:

RH1: The performance of XL-Lexeme in the Polish language will be comparable to that presented for other languages (except for Latin) in the original XL-Lexeme paper ([Cassotti et al., 2023](#)).

Based on the historical background of the two time periods investigated in this thesis, the LSC detected by XL-Lexeme in Polish will most probably concern political and technological terms. This is due to the 20th century being at the same time politically unstable and technologically progressive. For the same reason, all of the consequences of language change (widening, narrowing, amelioration and pejoration of meaning) presented by [Ullmann \(1951\)](#) and [Ullmann \(1962\)](#) are expected to be found. The hypothesis for RQ2 may therefore be summarised below:

RH2: LSC will most likely concern political and technological terms and all types of LSC consequences denoted by [Ullmann \(1951, 1962\)](#) will be observable.

2 Background

2.1 Distributional Hypothesis

The ideas behind modern LSC detection models originate from distributional semantics, which assumes that the meaning of a word can be induced by examining its distributional properties in large text corpora (Lenci & Sahlgren, 2023). The fundamentals of distributional semantics lie in the distributional hypothesis proposed by Harris (1954) and later popularised by Firth (1957) who famously claimed that “you shall know a word by the company it keeps” (Firth, 1957, p. 11). According to the distributional hypothesis, words that appear in similar contexts tend to have similar meanings (Lenci & Sahlgren, 2023). For example, both ‘roses’ and ‘tulips’ are equally likely to occur in the sentence “Please water my roses/tulips.” This implies that they are closely related in meaning, both referring to a kind of flower. What is more, the distributional hypothesis underpins many computational models of natural language processing (NLP), including word embedding techniques such as Word2Vec, GloVe, and FastText. More about word embeddings—a topic crucial to this thesis—can be found in subsection 2.4.

2.2 Neural Networks

Given that a transformer-based model serves as the cornerstone of this thesis, it is necessary to properly explain the concept of transformers. A simple way to introduce transformers is to see them as a specific type of neural network. Even those unfamiliar with computer science might recognize transformers from everyday applications like ChatGPT or Google’s Gemini, where they are used for language generation. Their relevance in the following thesis, however, lies in their capacity to encode contextualized word embeddings, as elaborated upon in subsection 2.3.

2.2.1 FNNs and RNNs

Understanding the transformer architecture requires a certain level of background knowledge on neural networks. The building blocks of every neural network are the so-called “neurons”. Those neurons, mimicking the human brain, are stacked into layers: the input layer, the hidden layer(s) and the output layer. Neurons from each of the neighbouring layers are connected with each other and each connection is characterised by a specific weight. The purpose of the weights is to determine the strength of the influence of one neuron on the other. Moreover, the output of each neuron is summed with a mathematical function called the “activation function”, thus making the model more precise and able to work with complex data (Saleem, 2023).

There exist two broad types of neural networks: a feedforward neural network (FNN) and a recurrent neural network (RNN). The principles behind their workings are very similar. As Pranshu (2024) notes, during the supervised training of FNNs, each neuron in the input layer receives some kind of data that is then passed through the hidden layers of the model, undergoing various transformations, which depend on the connection weights and the activation function. After the data travels through the hidden layers, its modified version is passed onto the output layers which produce results. The results produced by the model are then compared to the target values and a so-called loss function measures their discrepancy. The output of the loss function, called loss score, is then passed to an optimizer, which in turn adjusts the weights of the neural connections in such a way so as to minimize the difference between the model’s output and the target values.

As Poudel (2023) explains, RNNs’ architecture mimics that of FNNs with the addition of the so-called hidden states. Hidden states refer to the internal states of the neural network at any given time point. What is important is that they are influenced by the current input and the previous hidden state. In simpler terms, a hidden state is a kind of “memory box”, which updates with every new input, at the same time retaining

the previously processed information. This means that, while FNNs' information flow is uni-directional, RNNs process data recurrently. One of the first RNNs was introduced by [Elman \(1990\)](#), who presented his invention's ability to perform such tasks as sequence predictions. Soon enough, [Elman \(1990\)](#)'s model was followed by the invention of new RNN-based models, each improving on the original's deficiencies. Some of the most notable inventions include LSTM and BiLSTM models that account for the vanishing gradient problem.

2.2.2 Transformers

Although the first transformer, proposed in a revolutionary paper "Attention Is All You Need" ([Vaswani et al., 2017](#)), was built upon an encoder-decoder architecture, some of the subsequently invented models utilize encoder- (e.g. BERT) or decoder-only (e.g. GPT) architecture. For the sake of this thesis, which utilises XL-Lexeme—a model leveraging encoder-only architecture—only the encoder part of the original transformer will be explained using [Starmer \(2023\)](#)'s explanatory video as the supporting material.

The role of the encoder is to generate contextualised word embeddings out of the input text sequence. To better illustrate the information flow in the encoder, let us take an example sentence: "Dinosaurs eat humans", tokenised into entities "Dinosaurs", "eat", "humans", <EOS> (end of sequence token).

Each token of this sequence is connected to n number of activation functions and each connection gets assigned a random weight, which is later adjusted during the training process. The initial value assigned to each token is then multiplied by the connection weight and the resulting numbers are put into the activation functions.

At this point, word order is not accounted for. This means that the sentence could be interpreted as both "Dinosaurs eat humans" and "Humans eat dinosaurs". In response to this problem, positional encoding is applied. During positional encoding, each token gets assigned a positional vector that represents the token's position in the input sequence. This vector is then added to the outcome of the activation function, creating an embedding that is yet to be processed by the self-attention layer.

In general, self-attention works by comparing each token to every other token in the sentence, including itself. To do this, all of the position-encoded values in the token embedding are multiplied by their respective weights and their products are added together. This is done n times, resulting in n new values. The values obtained via these calculations are called "Query" values. The same process is repeated, again with a completely new set of weights, creating a set of values called "Key" values. The process leading to the creation of "Key" values is then repeated for all of the other tokens in the sequence.

Next, each of the n "Query" values is multiplied by its positional counterpart in the "Key" values for each token and the products of those multiplications for each "Query"- "Key" pair are then added, resulting in similarity scores between the token in question and all the other tokens in the sequence, including itself. Those similarity scores are then run through a SoftMax function, which gives out a number that determines what percentage of each token should be used to create an embedding for the token in question.

Next, the same process that created "Query" and "Key" values is repeated for each token, again with different weights, resulting in n values, cumulatively called "Values", for each token. Each of the "Values" values is then multiplied by its positional counterpart of SoftMax outputs. The resulting numbers for each token are then added together, creating the self-attention values for the token in question. Usually, encoders consist of multiple self-attention layers, which enables them to capture different relationships among the words. The self-attention values obtained from the self-attention layer are then added to the position-encoded values, creating residual connection values, which can be considered the final values that go into the word embedding.

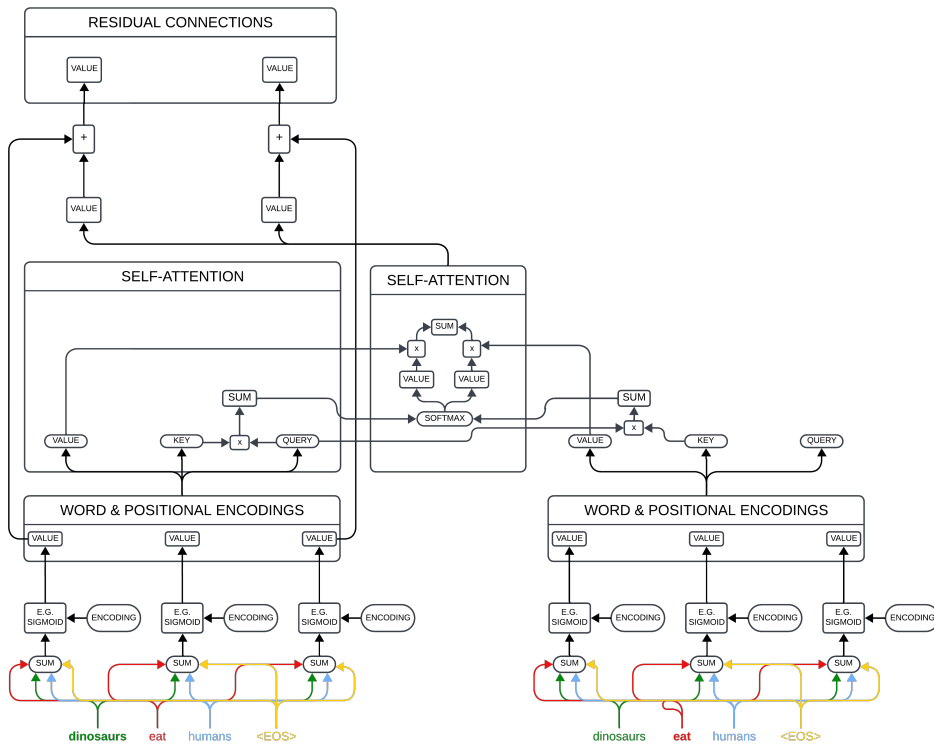


Figure 1: Simplified transformer architecture

2.3 Automatic Lexical Semantic Change Detection

While LSC researchers often opt for different models and techniques, the principles behind their LSC detection processes are usually quite similar. This subsection provides a brief guidebook to automatic LSC detection, walking the reader through the steps adopted by researchers employing LSC detection models and contributing articles to this field. Automatic LSC detection requires access to large text corpora divided into predefined time periods. These corpora are carefully preprocessed. This often involves tokenisation and lemmatisation. Next, word embeddings are created through the application of different models of meaning such as Word2Vec, GloVe or BERT etc. The choice of the model depends on the specific task requirements, data size and available resources. Despite the diversity of the models leveraged so far, it is difficult to determine the objectively best model for the LSC detection task. The obtained word embeddings are then compared between the chosen time periods with various models of change that employ distance metrics such as cosine distance. This way, significant shifts in semantics are detected.

2.4 Embeddings

To further illuminate the process of LSC detection, it is important to cover in more detail the concept of embeddings. In the realm of NLP, embeddings refer to computer-readable vector representations of textual entities. While embeddings can represent various text units, this subsection will focus exclusively on word embeddings. This decision stems from the topic of this thesis, which centres on lexical semantic change. Presently, there essentially exist three different methods of capturing meaning through embeddings: (1) count-based embeddings (e.g. TF-IDF) represent words based on their frequency of occurrence in a given corpus, (2) static embeddings are created via more sophisticated techniques such as global co-occurrence matrices (e.g. GloVe), and (3) contextual embeddings (e.g. BERT) are sensitive to the context in which words appear (Tahmasebi & Dubossarsky, 2023).

In modern Lexical Semantic Change detection tasks, count-based embeddings are rarely used due to their limited effectiveness. The crux of the debate lies in static (type) versus contextual (token) embeddings (Laicher et al., 2021). As Mei (2020) explains, type embeddings represent words as fixed vector representations that remain static and do not change based on the context of a word. He further notes that token embeddings, on the other hand, portray each instance of a word with a separate vector representation, taking into account the context of the word. LSC researchers often experiment with both kinds of embeddings, their final choice depending on the specific goals of their analysis. While type embeddings tend to be more efficient and simpler to analyse, token embeddings provide more detail and are able to detect subtler shifts in the meaning (Ehrmanntraut et al., 2021).

Word embeddings play a pivotal role in LSC detection, providing computers with readable word representations that can be easily compared through mathematical operations. If two word vectors occur close to each other in a vector space, their meanings tend to be similar. Lexical semantic change, on the other hand, occurs whenever the mathematical distance between two vector representations of the same word from different time periods is significantly high.

2.5 Literature Review

To this date, using word embeddings in computational LSC detection has become a standard approach. Naturally, this method is a relatively new invention, preceded by other techniques such as, for example, counting raw word frequencies (Hilpert & Gries, 2016).

The first word embeddings in LSC detection were usually type-based, created with the help of various algorithms that reduced their dimensionality while preserving semantic information. An early empirical study by Sagi et al. (2009) used the Latent Semantic Analysis (LSA) algorithm to computationally represent English words within specific time periods. LSA, a technique based on Singular Value Decomposition (SVD), constructs embeddings by analyzing the statistical co-occurrence patterns of words in a large corpus. Sagi et al. (2009) then calculated cosine similarity to determine the degree of lexical semantic change that occurred for each of the target words. The study by Sagi et al. (2009) served as a pioneer in the field of computational LSC detection and was soon followed by numerous new studies, each improving on the original results and some introducing new types of models and approaches. In the same year, Jurgens & Stevens (2009) created their embeddings using the Random Indexing (RI) algorithm, which operates on co-occurrence matrices to capture word associations. Two years later Gulordava & Baroni (2011) used sparse co-occurrence matrices that were weighted by Local Mutual Information—a method aimed at capturing semantic relatedness between words based on their local contexts.

Around 2014, the diversity of approaches to LSC began to increase (Kutuzov et al., 2018). One of the more progressive studies was conducted by Mitra et al. (2014), who used word clustering not only to detect LSC but also to determine its type. This involved grouping words with similar semantic properties into clusters, allowing for a more nuanced analysis of semantic shifts. Furthermore, Kim et al. (2014) pioneered the use of neural language models, specifically Continuous Skipgram with negative sampling (SGNS), to identify semantic shifts and pinpoint the specific time periods during which these shifts occurred. SGNS, a variant of the Skipgram model in neural network architectures, learns embeddings by predicting the context words given a target word. Hamilton et al. (2016) later evaluated SGNS and other embedding methods, concluding SGNS' superiority over distributional models based on Positive Pointwise Mutual Information (PPMI).

Until the late 2010s, automatic LSC detection was a topic covered by individual studies rather than taught about at larger-scale tutorials or workshops. Only in 2019, Eisenstein (2019) gave the first tutorial on the topic of LSC and the first international workshop on computational approaches to historical language change took place (Tahmasebi et al., 2019). The LSC detection task became more widely known in 2020 when it was established as the first task of the 14th International Workshop on Semantic Evaluation (Schlechtweg

et al., 2020). The organizers of the workshop noted that, at the time, the field lacked standard evaluation tasks and data, which was slowing down the development of new, finer models for LSC detection and depriving the researchers of comparable results that could definitively distinguish the superior models (Schlechtweg et al., 2020). SemEval2020 was thus groundbreaking—for the first time, the researchers were provided with an evaluation framework and manually annotated datasets relying on approximately 100,000 instances of human judgment. The shared task introduced by SemEval2020 consisted of two subtasks: (1) binary classification (predicting whether a change occurred or not) and (2) ranking a set of target words according to their degree of LSC. Four languages were studied by the participants: English, German, Latin and Swedish. For each language, two time-specific corpora were used, their spans differing between the languages. The works submitted for SemEval2020 utilised both type and token embeddings. According to the authors of the SemEval2020 paper, models leveraging type embeddings performed generally better than those using token embeddings. These results were further replicated by DIACR-Ita shared task on Italian LSC detection (Basile et al., 2020). The proven superiority of type embeddings caused a general surprise in the field, resulting in a number of papers investigating the issue of token embeddings and their performance in LSC detection. According to Laicher et al. (2021), the poorer performance of token embeddings stems from orthographic information on the target word, encoded even in the higher layers of BERT representations.

After the rise of transformers, many new LSC detection studies utilising models such as BERT or XML-RoBERTa came to rise. Only one year after the publication of SemEval2020 and DIACR-Ita results, Kutuzov & Pivovarova (2021) organised another shared task on LSC detection, this time pertaining to the Russian language (RuShiftEval). While RuShiftEval was, in many respects, similar to the previous shared tasks on LSC detection, several novel decisions were incorporated. Firstly, LSC detection was performed not on two, but three, time periods. Secondly, a training set was provided to the participating teams before they started developing their models. The results of RuShiftEval are groundbreaking in more than one way. For once, they proved that the use of a training set for training or fine-tuning the LSC detection models can lead to better model performance. Moreover, RuShiftEval was the first shared task in which token-based models (i.e. XLM-R, BERT and ELMo) clearly outperformed type-based models, taking the top five positions on the leaderboard. Finally, the best and the second best models in RuShiftEval relied on the XML-R model that was not specifically trained for Russian—it was trained in about 100 languages, including Russian. The submissions to the RuShiftEval task largely influenced the decisions made on the proceedings of the study presented in this thesis.

2.6 XL-Lexeme

WiC Pretrained Model for Cross-Lingual LEXical sEMantic change (XL-Lexeme, for short) is a model leveraged in this thesis (Cassotti et al., 2023). This subsection explains in more detail the workings of XL-Lexeme and the motivation behind its selection as the cornerstone of this study.

XL-Lexeme is built upon XLM-RoBERTa (XLM-R), a pre-trained multilingual transformer-based language model that produces contextualised (i.e. token) embeddings. As its official name suggests, XL-Lexeme is trained on the Word in Context (WiC) task, the aim of which is to determine whether or not the meaning of a target word is the same in the two given contexts. As Cassotti et al. (2023) claim, the idea to train an LSC model on the WiC task is grounded on the assumption that a model capable of detecting synchronic (i.e. co-occurring) changes will be equally robust at detecting diachronic (i.e. historical) changes. XL-Lexeme is a bi-encoder model leveraging a Siamese Network. In contrast to cross-encoders that encode two input sequences simultaneously, bi-encoders encode two sequences into two separate vectors and then compute their similarity (Mudadla, 2023). For many tasks, cross-encoders prove to be more accurate than bi-encoders, however, for tasks that require a distinct meaningful representation for each sentence (e.g. WiC), bi-encoders are more suitable (Cassotti et al., 2023).

The main advantage of XL-Lexeme, also noted by its creators, is the efficiency stemming from the utilisation of a bi-encoder, rather than the more computationally heavy cross-encoder. Considering this thesis’ limited access to computational power, the choice of XL-Lexeme appears adequate. Moreover, XL-Lexeme proves to be a state-of-the-art model, capable of handling multiple languages such as English, German, Swedish, and Russian. Especially important here is XL-Lexeme’s success in managing Russian, considering that it is closely related to the Polish language in terms of lexicon and grammar. Efficiency and multilingual data handling are therefore the main reasons for leveraging XL-Lexeme as the model of choice in this thesis.

2.7 Algorithms

Each target word, for which LSC is measured, is represented by multiple embeddings, which are, in turn, gathered into two matrices—one for time period t_1 and the second for time period t_2 . Since there are multiple embeddings per one word, measuring similarity between words from both time periods requires specific change detection algorithms. Following [Kutuzov & Giulianielli \(2020\)](#)’s paper, the algorithms chosen for this study are prototype distance (PRT) and average pairwise distance (APD). Their workings are described below.

PRT is an intuitive algorithm that averages all word embeddings in the matrix, resulting in a singular prototype word embedding. After creating prototype embeddings for both time periods, a similarity measure such as cosine similarity can be applied. The visualisation of the algorithm can be seen in figure 2. The operation for PRT, that produces cosine distance d , is:

$$\text{PRT}(U_w^{t_1}, U_w^{t_2}) = d\left(\frac{\sum_{x_i \in U_w^{t_1}} x_i}{N_w^{t_1}}, \frac{\sum_{x_j \in U_w^{t_2}} x_j}{N_w^{t_2}}\right)$$

where $U_w^{t_1}$ and $U_w^{t_2}$ are the matrices of w in time periods t_1 and t_2 , and $N_w^{t_1}$ and $N_w^{t_2}$ are the number of occurrences of w in both time periods.

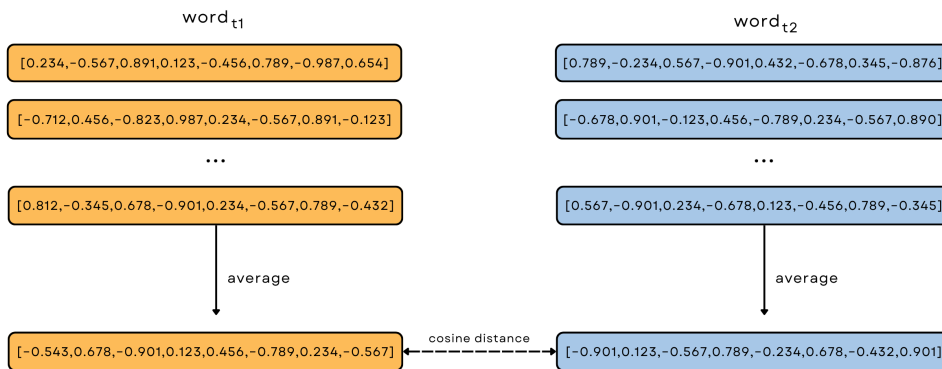


Figure 2: Prototype Distance (PRT)

With APD, the change is measured by calculating the average cosine distance between all possible pairs of embeddings between t_1 and t_2 . The visualisation of APD can be seen in figure 3. The operation that calculates APD is:

$$\text{APD}(U_w^{t_1}, U_w^{t_2}) = \frac{\sum_{x_i \in U_w^{t_1}, x_j \in U_w^{t_2}} d(x_1, x_2)}{N_w^{t_1} N_w^{t_2}}$$

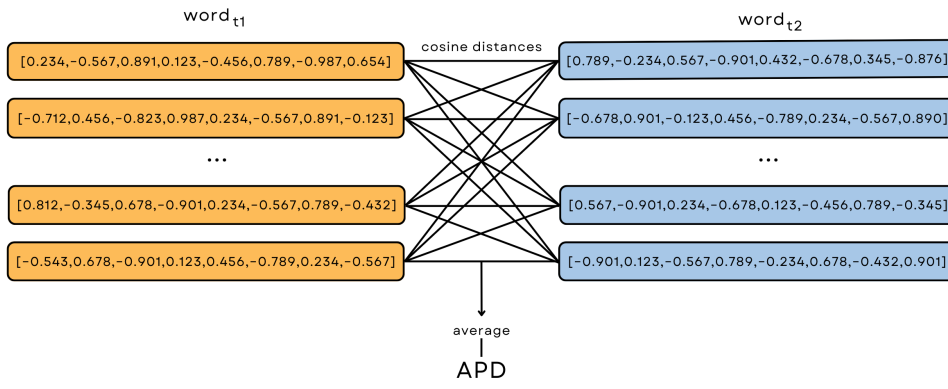


Figure 3: Average Pairwise Distance (APD)

3 Experimental Setup

3.1 Dataset

The search for an adequate data source for the thesis was conducted on CLARIN-PL, a Polish branch of Common Language Resources and Technology Infrastructure. CLARIN-PL is a scientific consortium where electronic language resources are linked. Among the resources freely provided by CLARIN-PL, the Polish Parliamentary Corpus (PPC) emerged as the most exhaustive source of textual data.

3.1.1 Motivation

The reasons for choosing the PPC as the cornerstone of this study are manifold. Firstly, out of the few publicly available Polish corpora, the PPC emerges as the most comprehensive source of textual data. Moreover, as the PPC reports on natural speech rather than scripted prose, the corpus gives insights into the linguistic changes that have been occurring in real time. Due to the formal context of the transcribed speeches, the PPC proves to be suitable for LSC in a controlled environment. What is more, parliamentary speeches cover a wide range of topics, including societal, cultural, and political changes which can be directly tied to the semantic shifts in language. Lastly, as was mentioned in subsection 1.5, the PPC spans from the 1910s to the 2020s—years filled with major historical events that could have induced a number of semantic changes.

3.1.2 Properties

The Polish Parliamentary Corpus is a collection of transcribed plenary sittings of the Sejm and Senate (lower and upper houses of the parliament, respectively) of the Polish Republic dating from 1919 to 2023. It includes transcripts of Sejm sittings, Sejm committee sittings from 1993, Sejm interpellations and questions from 1997, Senate sittings from 1922–1939 and 1989 to present and Senate committee sittings from 2015 (CLARIN-PL, 2023). The PPC amounts to over 340 thousand documents and almost 750 million tokens. Both linguistically annotated and unannotated versions of the corpus are available.

3.1.3 Preprocessing

For the purpose of this study, a linguistically annotated version of the PPC was utilised. Two types of XML documents were used for each plenary sitting: (1) one consisting of utterance-level segmentation, tokenisation and lemmatisation produced with Morfeusz2 (ann_morphosyntax), and (2) one with disambiguated morphosyntactic description produced with Concraft2 (text_structure). Before using the PPC data, two preprocessing steps were applied to the dataset. Firstly, documents spanning from 1962 to 1988 were removed, dividing the dataset into two distinct time periods: 1919-1961 (t_1) and 1989-2023 (t_2). At this point, t_1 consisted of 1269 parliamentary sitting transcriptions and t_2 of 471102. Next, to balance the number of transcriptions between the two time periods and shorten the processing time of the files, t_2 was cut down to 2130 transcriptions by keeping only the ones from Sejm plenary sittings. The final token and lemma counts can be seen in table 1.

Table 1: Final token and lemma counts

time period	tokens	lemmas
t_1	26,110,768	101,151
t_2	134,202,395	141,669

3.2 Extracting Data

The process of extracting and formatting data for XL-Lexeme’s input involved several steps, each described in its separate subsection ¹.

3.2.1 Lemmas and Word Forms

The first step consisted of extracting lemmas and word forms from `ann_morphosyntax` files using regular expressions. Specifically, a list of tuples was created for each time period, where each tuple consisted of four items: (1) lemma, (2) word form, (3) number of the segment in which the target word occurs, and (4) path to the currently used `ann_morphosyntax` file containing the data. An example tuple from the resulting list would look as follows:

```
('sprawozdanie', 'Sprawozdanie', 'u-1.0', 'files/C1/1919-1922/sejm/posiedzenia/pp/191922-sjm-ppxxx-00001-01/ann\_morphosyntax.xml')
```

To better illustrate the extraction procedure, an excerpt from a random `ann_morphosyntax` file can be found below. The lemma was extracted from the `<f name='base'><string>lemma</string></f>` tag (line 9), the word form was extracted from the `<f name='orth'><string>word</string></f>` tag (line 4) and the segment number was extracted from the `<seg corresp='segment' xml:id='segment'>` tag (line 1).

```
1. <seg corresp='ann_segmentation.xml#segm_u-1.0.1-seg' xml:id='morph_u-1.0.1-seg'>
2.   <fs type='morph'>
3.     <f name='orth'>
4.       <string>Sprawozdanie</string>
5.     </f>
6.     <f name='interps'>
7.       <fs type='lex' xml:id='morph_1.1.1.1.1-lex'>
8.         <f name='base'>
9.           <string>sprawozdanie</string>
10.        </f>
11.        <f name='ctag'>
12.          <symbol value='subst'/'>
13.        </f>
14.        <f name='msd'>
15.          <symbol value='sg:nom:n:ncol' xml:id='morph_1.1.1.1.1-msd'/'>
16.        </f>
17.      </fs>
18.    </f>
19.    <f name='disamb'>
20.      <fs type='tool_report'>
21.        <f fVal='#morph_1.1.1.1.1-msd' name='choice'/'>
22.        <f name='interpretation'>
23.          <string>sprawozdanie:subst:sg:nom:n:ncol</string>
24.        </f>
25.      </fs>
26.    </f>
27.  </fs>
28. </seg>
```

3.2.2 Comparing Lemmas

The next step involved comparing lemmas from both time periods to generate a dictionary of common lemmas (`common_dictionary`). First, the lists from t_1 and t_2 were turned into dictionaries, where the lemma was the key, while the value was a list of tuples consisting of (1) word form, (2) segment number, and (3) file path. An example item from this dictionary is given below:

```
'sprawozdanie': ('Sprawozdanie', 'u-1.0', 'files/C1/1919-1922/sejm/posiedzenia/pp/191922-sjm-ppxxx-00001-01/ann\_morphosyntax.xml')
```

¹All code is available on GitHub <https://github.com/ewaslowinska/LSC-in-Polish-using-XL-Lexeme>

By counting the length of each of the lists, the frequency of occurrence of each lemma in both time periods was counted and additional dictionaries for t_1 and t_2 were created where the key was the lemma and the value was the number of its occurrences. The first three items from t_1 can be seen below:

```
{'sprawozdanie': 14877, 'stenograficzny': 1333, 'z': 411342}
```

By comparing both of the newly created dictionaries, yet another dictionary was created. The new dictionary contained lemmas shared between both time periods as its keys and tuples of (1) frequency in t_1 and (2) frequency in t_2 as its values. An excerpt from this dictionary is shown below:

```
{'pniak': (11, 2), 'zaprzyjaźniony': (136, 239), 'naocznie': (90, 88)}
```

3.2.3 Final Lemma List

Further, a final list of lemmas to be examined by the model was created. The dictionary resulting from subsection 3.2.2 was cleaned from lemmas that had fewer than 25 instances in either of the time periods due to insufficient data to yield meaningful results. Furthermore, the mean occurrence frequency of the remaining lemmas across both time periods was calculated and a new dictionary was created where lemmas were the keys and mean occurrence frequencies were the values. The dictionary was sorted from the highest to the lowest value and the first 2000 lemmas from the new dictionary were then removed assuming them to be primarily function words where LSC would not be observed. This left the dictionary with a lemma count of 12869.

3.2.4 Random Sentence Selection

For each lemma from the dictionary resulting from subsection 3.2.3, 25 sentences per time period were randomly selected. This was done by extracting file paths from the dictionary created in subsection 3.2.2.

3.2.5 Sentence Extraction

After opening the right text_structure file, the utterance of the right number (containing the target word form) was found using regular expressions. To better illustrate the file structure, an extract from a random text_structure file can be seen below:

```
<u xml:id='u-1.0' who='#komentarz'>Sprawozdanie stenograficzne z 1. posiedzenia Sejmu Ustawodawczego</u>  
<u xml:id='u-1.1' who='#komentarz'>z dnia 10. lutego 1919 r.</u>  
<u xml:id='u-1.2' who='#komentarz'>(Początek posiedzenia o godzinie 11 minut 45.)</u>
```

There were instances where the utterance ended in the middle of the sentence, which was then picked up in the following utterance. This was accounted for using boolean values which indicated whether the utterance ended on either one of (', '!', '?') or not. If the utterance did not end on either of them, the next utterance was also scanned until one of these punctuation marks was found. Sometimes the text obtained that way would consist of more than one sentence, in which case regular expressions were, once again, utilised to find the sentence in which the target word occurs. After that, the results were gathered in a dictionary with key-value pairs denoting: (1) sentence, (2) index of the first letter of the target word, (3) index of the last letter of the target word, and (4) lemma. An example dictionary would be:

```
{'sentence': 'Ponad 25 abonentów radiowych i telewizyjnych.', 'first_index': 9, 'last_index': 17, 'lemma': 'abonent'}
```

Next, a JSONL file for each lemma was created per time period, containing the dictionaries for each sentence in which this lemma occurs.

3.3 LSC Detection

After obtaining the data required for XL-Lexeme, the model could be utilised, taking sentences and lemma indices as its input and outputting token embeddings. The token embeddings for each lemma in both time periods were then gathered in embedding matrices. Since 25 sentences per time period were randomly selected in 3.2.4, the matrices should have consisted of 25 embeddings each, however, due to problems with sentence extraction described in subsection 3.2.5, some matrices consisted of fewer embeddings. In t_1 , the least amount of embeddings in a matrix was 20 and the number of matrices with less than 25 embeddings was 528, constituting 4% of the total number of matrices. In t_2 , the minimum number of embeddings in a matrix was 22 and there were 251 matrices with less than 25 embeddings, which amounted to 2% of all matrices. The algorithms discussed in subsection 2.7 were then applied.

3.4 Analysis

To analyse the obtained results and judge XL-Lexeme’s performance, a qualitative analysis of the LSC detection results and a WiC task are performed.

3.4.1 Qualitative Analysis (LSC)

Running word embeddings through PRT and APD yielded two lists of lemmas sorted from the highest cosine distance to the lowest. For the purpose of the qualitative analysis, both lists were shortened to their respective top 100 lemmas with the highest cosine distance (the lists can be found in appendices A.1 and A.2). For each of the 100 lemmas, the sentences in which they appeared were analysed through close-reading to see whether the LSC detected by the model actually took place and what, if any, types of changes could be observed. This was done by noting down the meaning of the target lemma in each of the analysed sentences, summing up the frequencies of each of the observed meanings and comparing those meanings and their frequencies between the two time periods.

3.4.2 WiC Task

Considering that XL-Lexeme has never been used for LSC detection in the Polish language, its performance in this study remains doubtful until it is measured quantitatively. As was mentioned in subsection 2.6, XL-Lexeme is a WiC model, working on the assumption that synchronic change detection functions on the same basis as diachronic change detection. In other words, no matter the time span between the two words in context, XL-Lexeme should create similar embeddings for the words with the same meaning and significantly different embeddings for the words with different meanings.

To test this WiC-solving ability of XL-Lexeme in the context of the unexamined Polish language, manual annotation of a subset of t_1 - t_2 sentence pairs was performed by two human annotators. Both annotators were native Polish speakers with an undergraduate-level diploma in linguistics. A union of the top 50 most changed lemmas from PRT- and APD-yielded lists was utilised. If a certain lemma was repeated in PRT- and APD-yielded lists, a new lemma from the PRT-yielded list was chosen. For each lemma, 10 pairs of t_1 - t_2 random sentences from the PPC were extracted, resulting in a starting list of 1000 sentence pairs to annotate. This number, however, shrank during the annotation process, as some sentences were discarded by the annotators, for example, due to OCR errors in the target lemma. The remaining sentence pairs were classified into binary labels, based on whether the target word meaning was different between them (0 for same, 1 for different). The final annotation scores consisted of an average score given by the two annotators.

If the annotation differed between the annotators, the sentence pair was discarded. Interannotator agreement (Cohen's Kappa) was calculated.

Each sentence from the annotation dataset was processed by XL-Lexeme to create a target word embedding. The embeddings for the target words in the sentence pairs were then compared using cosine distance. A threshold of 0.5 was set to classify the resulting scores as different or the same. These automatically generated labels were then compared against the manual annotations and the accuracy, precision, recall and F1-score were measured.

4 Results and Discussion

4.1 Numerical Results

In the lists of the top 100 changed words according to PRT and APD, 46 words repeat. Overall, PRT tends to yield lower cosine distances than APD, their average scores being 0.0294 and 0.587, respectively. The highest cosine distance obtained through PRT equals 0.569, while for APD it equals 0.813. The most changed word according to both algorithms appears to be *oświecić* ('to explain' → 'to light up'). The lowest cosine distance, implying the least changed word, was: for PRT, *wdzięcznić* ('to simper') with the score 0.000781, and for APD it was *Wróbel* (surname), which obtained the score of 0.427. These results are summed up in table 2.

Table 2: Numerical results yielded by PRT and APD

	PRT	APD
highest cosine distance	<i>oświecić</i> 'to explain/to light up' = 0.569	<i>oświecić</i> 'to explain/to light up' = 0.813
lowest cosine distance	<i>wdzięcznić</i> 'to simper' = 0.000781	<i>Wróbel</i> (surname) = 0.427
mean cosine distance	0.0294	0.587

4.2 WiC Task

4.2.1 Annotation

After discarding the unusable sentence pairs, annotators were left with 866 sentence pairs to annotate, amounting to 88 distinct lemmas. After checking for agreement between the annotations, 133 sentence pairs had to be further eliminated, resulting in 733 sentence pairs.

Cohen's Kappa was calculated and equalled 0.585, indicating a moderate interannotator agreement. For a subjective task like this, however, an agreement of around 0.6 is a rather good result—annotating a corpus for a WiC task often depends on an individual's language instinct and understanding of the target word. Moreover, when annotating on a binary scale, the annotators often disagree due to their varying perceptions of the target word's meaning's level of difference in the given contexts.

Two main types of disagreements could be found in the annotations. One common type of disagreement concerns the use of the target word in a drastically different context but modifying this context in a similar manner. This can be illustrated on the example:

Tworzy się coś w rodzaju uniwersytetu korespondencyjnego.

'A kind of **correspondence** university is being created.'

↓

Jedną z metod dotarcia kandydata do wyborcy jest metoda korespondencyjna [...].

'One method for a candidate to reach a voter is the **correspondence** method [...].'

In this example, *korespondencyjny* ('correspondence') refers either to the type of studies or a method of communication. This difference in contexts caused one of the annotators to mark the meaning of the

target words in both sentences as different. The second annotator disagreed, however, as they noticed that “correspondence studies” refer to a method of studying which involves correspondence by letter, which is fairly similar to what is meant by *korespondencyjny* (‘correspondence’) in the second sentence.

Another type of disagreement often occurred in cases where the use of the target word was metaphorical in one context and literal in the other. One of the annotators was more inclined to classify such cases as having different meanings, while the other annotator treated them as the same. This can be seen in the example below.

[...] żyje jeszcze jad **wszczepiony** przez okupanta [...].

[...] the venom **implanted** by the occupying forces is still alive [...].

↓

[...] użyta wobec osób czy to po zawale serca, czy z **wszczepionym** rozrusznikiem [...].

[...] used against people whether after a heart attack or with an **implanted** pacemaker [...].

As we can see, in the first sentence, the word *wszczepić* (‘to implant’) is used in a metaphorical way to signify something being integrated into the minds of people. In the second sentence, however, the meaning is rather literal, concerning the physical implanting of foreign matter into the human body.

4.2.2 Results of the WiC Task

According to the ground truth labels obtained via manual annotation described in subsection 3.4.2, in 588 sentence pairs the meaning of a target lemma was different, while in 145 it was the same. This suggests a significant, fourfold dominance of true positive labels over true negative labels.

The ground truth and predicted labels were compared using measures such as accuracy, precision, recall and F1-score. These results can be seen in table 3.

Table 3: Accuracy, precision, recall and F1-score for the WiC task

Accuracy	Precision	Recall	F1-score
0.730	0.971	0.684	0.802

The accuracy of 0.730 indicates a rather robust performance of XL-Lexeme on the WiC task in the Polish language. From the precision score of 0.971 in table as well as in the confusion matrix in figure 4, we can see that the model is very reliable when it comes to identifying difference in meaning. In other words, the high precision indicates that whenever the model classifies the target lemma’s meanings as different, more than 97% of the time, it is right. A lower recall score of 0.684, however, indicates that the model recognises differences only 68% of the time and often labels different meanings as the same. Since the data is rather unbalanced, with true positives being far more frequent than true negatives, F1-score of 0.802 gives a better understanding of the model’s performance than accuracy. In summary, while the model performs fairly well on the WiC task, there is still room for improvement, especially when it comes to recall.

The results of this WiC task are on par with the results presented by the previous studies on LSC detection, including those competing in the SemEval2020 Task 1 Subtask 1 (Schlechtweg et al., 2020). The best

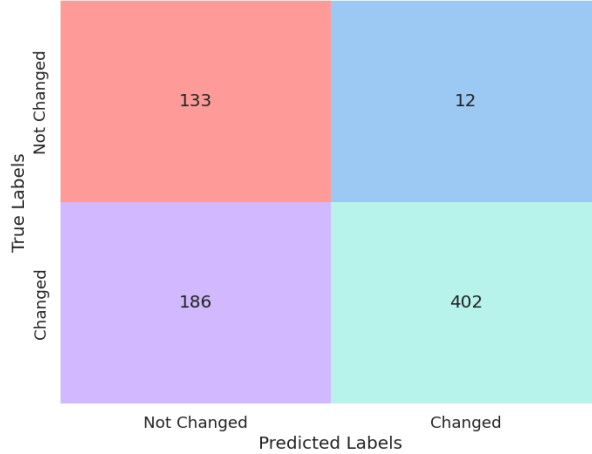


Figure 4: Confusion matrix for the WiC task

accuracy achieved in this task for the Swedish dataset, submitted by the Random team (Cassotti et al., 2020) amounted to 0.774. This result, however, was obtained through the use of type-based embeddings. When it comes to the best accuracy achieved by token-based models in SemEval2020, it equaled 0.710 and was obtained for Swedish dataset by DCC team (Zamora-Reina & Bravo-Marquez, 2020) and UiO-UvA team (Kutuzov & Giulianelli, 2020). By comparing those results to the accuracy obtained in this study, one can see that XL-Lexeme’s performance in Polish WiC task is on par with the best LSC models for Swedish. When it comes to other languages studied in SemEval2020—English, German and Latin—the best scoring models, accuracy-wise, usually fall behind XL-Lexeme’s performance in the Polish WiC task. However, in the context of this study, it is worth focusing on the F1-score rather than accuracy due to the unbalanced nature of the leveraged dataset. At SemEval2020, the highest noted F1-score was obtained by Zhou & Li (2020) for the Latin dataset and amounted to 0.769. Again, it is worth noting that Zhou & Li (2020) used type, rather than token, embeddings. The best F1-score obtained by a team utilising token embeddings equaled 0.767 and concerned the Latin dataset (Karnysheva & Schwarz, 2020). As it turns out, the F1-score of 0.802, obtained in the current Polish-based WiC task, is higher than any of the F1-scores reported by the participants of the SemEval2020. Also precision in this study proves to be ahead of those presented by majority of the SemEval2020 submissions for all the languages in question. Only recall-wise, XL-Lexeme performs moderately in Polish in comparison to other models and languages presented in SemEval2020.

(A)							(B)														
Team	Subtask 1					System	Avg.	English			German			Latin			Swedish				
	Avg.	EN	DE	LA	SV			P	R	F1	P	R	F1	P	R	F1	P	R	F1		
UWB	.687	.622	.750	.700	.677	type	.583	.789	.646	.6	.562	.58	.583	.824	.683	.769	.769	.769	.381	1.0	.552
Life-Language	.686	.703	.750	.550	.742	type	.549	.8	.63	.5	.625	.556	.441	.882	.588	.783	.692	.735	.471	1.0	.64
Jiaxin & Jinan	.665	.649	.729	.700	.581	type	.618	.672	.629	.556	.625	.588	.609	.824	.7	.889	.615	.727	.417	.625	.5
RPI-Trust	.660	.649	.750	.500	.742	type	.562	.7	.607	.5	.562	.529	.583	.824	.683	.7	.538	.608	.467	.875	.609
UG_Student_Intern	.639	.568	.729	.550	.710	type	.426	.971	.579	.432	1.0	.603	.366	.882	.517	.65	1.0	.788	.258	1.0	.41
DCC	.637	.649	.667	.525	.710	type	.523	.654	.564	.474	.562	.514	.542	.765	.634	.7	.538	.608	.375	.75	.5
NLP@IDSIA	.637	.622	.625	.625	.677	token	.512	.701	.56	.452	.875	.596	.395	.882	.546	.647	.423	.512	.556	.625	.588
JCT	.636	.649	.688	.500	.710	type	.683	.487	.559	.778	.438	.56	.667	.588	.625	.786	.423	.55	.5	.5	.5
Skurt	.629	.568	.562	.675	.710	token	.504	.567	.52	.667	.75	.706	.381	.471	.421	.611	.423	.5	.357	.625	.454
Discovery_Team	.621	.568	.688	.550	.677	ens.	.593	.499	.51	.5	.438	.467	.556	.588	.572	.9	.346	.5	.417	.625	.5
Count Bas.	.613	.595	.688	.525	.645	-	.5	.496	.493	.471	.5	.485	.5	.647	.564	.6	.462	.522	.429	.375	.4
TUE	.612	.568	.583	.650	.645	token	.625	.451	.492	.714	.312	.434	.524	.647	.579	.818	.346	.486	.444	.5	.47
Entity	.599	.676	.667	.475	.581	type	.622	.495	.478	.833	.312	.454	.524	.647	.579	.778	.269	.4	.353	.75	.48
IMS	.598	.541	.688	.550	.613	type	.468	.466	.464	.538	.438	.483	.267	.235	.25	.667	.692	.679	.4	.5	.444
cs2020	.587	.595	.500	.575	.677	token	.475	.462	.429	.462	.375	.414	.37	.588	.454	.833	.385	.527	.235	.5	.32
UiO-UvA	.587	.541	.646	.450	.710	token	.571	.361	.42	.75	.188	.301	.462	.353	.4	.739	.654	.694	.333	.25	.286
NLPCR	.584	.730	.542	.450	.613	-	.789	.243	.365	1.0	.188	.316	.857	.353	.5	.8	.308	.445	.5	.125	.2
Maj. Bas.	.576	.568	.646	.350	.742	token	.51	.481	.362	.75	.188	.301	.4	.824	.539	.5	.038	.071	.389	.875	.539
cbk	.554	.568	.625	.475	.548	token	.767	.226	.337	1.0	.188	.316	.667	.235	.348	1.0	.231	.375	.4	.25	.308
Random	.554	.486	.479	.475	.774	type	.682	.238	.318	1.0	.062	.117	.625	.294	.4	.818	.346	.486	.286	.25	.267
UoB	.526	.568	.479	.575	.484	topic	.437	.216	-	.5	.125	.2	.471	.471	.471	.778	.269	.4	0	0	-
UCD	.521	.622	.500	.350	.613	graph	-	.401	-	-	0	-	.346	.529	.418	.714	.577	.638	.25	.5	.333
RJIP	.511	.541	.500	.550	.452	type	-	.341	-	-	0	-	.4	.353	.375	.676	.885	.767	.2	.125	.154
Freq. Bas.	.439	.432	.417	.650	.258	-	-	.0	-	-	0	-	-	.0	-	-	.0	-	-	0	-

Figure 5: (A) = Accuracy scores obtained by the participating teams in SemEval2020 Task 1 Subtask 1, (B) = Precision (P), Recall (R) and (F1) F1-scores obtained by the participating teams in SemEval2020 Task 1 Subtask 1 (Schlechtweg et al., 2020)

The question remains—why does XL-Lexeme perform so well in the Polish WiC task? One of the possible reasons is that the base model behind XL-Lexeme, that is XLM-RoBERTa, has been trained in Polish alongside other languages. As [Conneau et al. \(2020\)](#) notes, XML-R has been trained on 6.49 billion Polish tokens, constituting 44.6 GiB of data. This amount of training data in Polish might have, figuratively, “accustomed” XML-R to the Polish language. Another reason lies in the fact that XL-Lexeme is a model that has utilised multiple evaluation datasets during its training on the WiC task. Some of those datasets, like e.g. AM2iCO ([Liu et al., 2021](#)), include languages of the same family as Polish (Russian, Bulgarian etc.). XL-Lexeme might have learned the rules governing such languages and transposed them onto the WiC task presented in this study.

4.3 Qualitative Analysis (LSC)

Close-reading of the 25 randomly chosen sentences (subsection 3.2.4) containing the top 100 words with the highest LSC score according to both algorithms confirmed the predictive abilities of the chosen methodology. All lemmas distinguished by the algorithms’ top 100 lists have undergone semantic changes of some kind. The types of changes vary and can be summed up as follows:

1. Acquisition of meaning - rarely a phenomenon by itself, usually took place simultaneously with loss or dissipation of certain meaning(s). An example of meaning acquisition can be seen in the word *regeneracja*, whose old meaning ‘recuperation’ has been complemented by the new meaning ‘recycling’, as presented in the sentences below.

*Ideaty [...] staną się podstawowym czynnikiem **regeneracji** i uzdrowienia.*

‘The ideals [...] will become a fundamental factor of **recuperation** and healing.’

↓

*Spółka [...] zajmuje się **regeneracją** części zamiennych [...].*

‘The company [...] is engaged in the **recycling** of spare parts [...].’

2. Loss of meaning - again, rarely a phenomenon by itself, usually accompanied by acquisition or popularisation of certain meaning(s). Meaning loss can be seen on the example of the word *oświecić*, where old meaning ‘to explain’ has been completely overtaken by the meaning ‘to light up’.

*Sprawozdanie to jest dorobkiem ścisłej i wyężonej pracy wszystkich członków komisji, którzy [...] postarali się wszechstronnie zgłębić i **oświecić** zagadnienie.*

‘The report is the result of close and hard work by all members of the committee, who tried to comprehensively explore and **explain** the issue [...].’

↓

*[...] niebo nad Puszcą Rudnicką **oświeciły** luny płonących domostw [...].*

‘[...] the sky above the Rudnicka Forest was **lighted** by the glow of burning houses [...].’

3. Change in meaning distribution - the most common type of LSC found in the PPC dataset, consisting of meaning popularisation and/or dissipation. It can be observed on the word *przeszczepić*, where meaning ‘to adopt’ started dissipating, while the meaning ‘to transplant’ was popularised.

[..] *Rząd jest zdecydowany najsurowiej wystąpić przeciwko tym, którzy chcą **przeszczepić** metody obce* [..].

‘[..] the Government is determined to act most severely against those who want to **adopt** foreign methods [..].’

↓

[...] *świadomość tego, że jej narządy można wykorzystać, można **przeszczepić** [...].*

‘[...] the awareness that their organs can be used, can be **transplanted** [...].’

For some of the semantic changes, it can be clearly said that historical or technological advancements were their direct causes. This can be seen in the word *wszczepiać* whose meaning changed from ‘to instil’ to ‘to implant’ due to the progress in medical technology. Some changes stem from the use of an old meaning in a metaphorical way or the other way around. For example, in the word *rozjeżdżać*, the old meaning ‘to sightsee (by going in many different directions)’ was replaced by the meaning ‘to get misaligned’. This can be seen as going from literal to metaphoric meaning—physically going in different directions turned into ideas drifting apart.

4.4 Noise

By close-reading the sentences containing the target words, it was possible to detect noise in the data, the effects of which can be seen in the results. Three main types of noise appeared in the data:

1. Optical Character Recognition (OCR) errors - the text of the PPC was obtained by passing image-based PDF files through FineReader OCR tool (Ogrodniczuk, 2018). Although the resulting text was verified by human proofreaders, OCR errors, such as the presence of a wrong letter or the absence of a certain letter, could still be found in some portions of the annotated corpus. Cases, where the OCR error was present in only one time period, resulted in certain lemmas scoring high on the distance measure. One example is when the word *poradzić* ‘to advise’ was incorrectly recognised as *porazić* ‘to electrocute’ in t_1 . Since *porazić*, with its correct meaning, was present in t_2 , the obviously different contexts of this lemma in t_1 and t_2 resulted in the cosine distance being high and placed *porazić* high on the PRT- and APD-yielded top 100 lists.
2. Non-words classified as words - for reasons such as the lack of proper expanding of abbreviations in the annotated corpus, some non-words have been incorrectly classified as words and further processed by the models of meaning and change, often yielding high cosine distance scores. While sometimes the non-words turned out to be gibberish (e.g. *ii* or *ki*), there were times when a non-expanded abbreviation had the same form as a real word. For example the abbreviation *gen.* (from *general* ‘general’) has the same form as the word *gen* ‘gene’. Since the ‘general’ abbreviation was utilised in t_1 while the ‘gene’ word was present in t_2 , the misclassified word *gen* scored high on the list of cosine distances.
3. Lemmatisation errors - the annotated corpus is characterised by frequent lemmatisation errors. Those errors stem mostly from the fact that the declination and inflection of certain lemmas cause their word forms to be identical to the declined or inflected word forms of other lemmas. This is, for example, the case for the word *zubożać* ‘to impoverish’, whose form *zubożali* ‘they impoverished’ is the same as in *zubożali* ‘they became poor’ stemming from the word *zubożeć* ‘to become poor’. As one can see, although both lemmas are very similar in their form, their meaning is fundamentally different. Mistaking *zubożeć* for *zubożać* in t_2 caused the cosine distance to be high, although no LSC had actually taken place.

Moreover, it is worth noting that the lemmatisation can be quite problematic in itself since no official guidelines are available. It is apparent that Morfeusz2, used for lemmatisation of the the PPC, classifies many word forms as one lemma, which often goes against the intuition of a native speaker. For example, it often lemmatises nouns as verbs from which they are derived. For instance, the word *tykanie* ‘ticking’ is lemmatised by Morfeusz2 as *tykać* ‘to tick’, although different lemmatisers, such as *trankit*, would treat it as a lemma in itself. Lemmatising derivatives as their root words might be problematic as it can potentially hinder the LSC detection. For example, *tykanie* might change in meaning while the meaning of *tykać* may remain the same.

4.5 Influence of Context

When concluding the qualitative analysis, it is necessary to consider the extent to which context influences the results. In other words, it is worth asking the question of whether it is possible that the changing context of utterances in t_1 and t_2 incorrectly suggests that LSC took place for words where no actual LSC occurred.

To answer this question, further analysis was performed on the list of most changed words according to PRT and APD. This part of the analysis consisted of examining the word meanings listed in two dictionaries: (1) for t_1 , a dictionary of the Polish language edited by Doroszewski (SJPD) and published through the years 1958-1969 was used, (2) for t_2 , modern-day electronic giant dictionary of Polish language (WSJP) published from 2007 to present was used. By comparing the words that have gained or lost their meanings according to the previously conducted analysis to the meanings listed in both dictionaries, it was possible to see whether the supposed gain/loss actually took place. The logic behind this comparison was that if a certain meaning is present in both dictionaries, it could not have been introduced only in t_2 nor could it have been lost.

In practice, it is difficult to determine whether the meaning was lost in t_2 even when looking into modern dictionaries. This is because even the most recently published dictionaries tend to contain word meanings which are no longer in use and oftentimes they do not state whether a certain meaning is outdated or not. This is why this part of the analysis focused more on the supposedly gained meanings by checking for their presence in the older SJPD dictionary.

Interestingly, the vast majority of the allegedly gained meanings turn out to be already present in the SJPD dictionary. Out of the set of top 20 most changed words according to PRT and APD, only four of the supposedly acquired meanings did not appear in the SJPD dictionary. This suggests a significant influence of context on the detection of LSC in this study. This is not surprising, considering the nature of parliamentary speeches. In a parliamentary context, only the most pressing issues are the topic of discussion, which may limit the use of a given word to only one of its meanings. For example, the word *wszczepiać* whose meaning ‘to implant’ has not appeared in t_1 and thus was labelled as a newly acquired meaning, actually appears in the old dictionary. The reason ‘to implant’ meaning was not present in t_1 and only appeared in t_2 was that no pressing issues regarding the medical implanting of certain substances or machines were present in t_1 and thus the parliament did not discuss them. On the other hand, the popularisation of the in-vitro method of conception in relatively recent years created many heated discussions in the parliament of t_2 , causing an explosion in the use of ‘to implant’ meaning of *wszczepiać*. This significant influence of context in the current study of LSC will be further discussed in the critiques and limitations section.

4.6 Answers to the Research Questions

At this point, it is possible to answer both research questions and check the truthfulness of their respective hypotheses. As the qualitative analysis of the LSC detection results suggests, the performance of XL-Lexeme in the Polish LSC detection task is rather robust. XL-Lexeme also proved to be a good model of

meaning for the Polish WiC task, which is strongly connected to the LSC detection ability of the model (as both LSC and WiC revolve around the detection of semantic difference between the target words). Especially admirable is the model's high precision in this task—XL-Lexeme is almost always right when classifying word meanings as different. Both WiC task's results and qualitative analysis of the LSC detection results therefore, agree on the point that XL-Lexeme's performance in the Polish language is very satisfactory and comparable to [Cassotti et al. \(2023\)](#)'s results. The hypothesis given to the first research question: "The performance of XL-Lexeme in the Polish language will be comparable to that presented for other languages (except for Latin) in the original XL-Lexeme paper ([Cassotti et al., 2023](#))" was therefore accurate.

The second research question considered the types of change that have occurred between t_1 and t_2 . As was discussed in subsection 4.3, every possible type of change can be found in these time periods—semantic widening and narrowing as well as the change in the meaning distribution, amelioration and pejoration. Some changes visibly stem from technological and political advancements, while the source of others is less apparent. Once again, the hypothesis to this research question: "LSC will most likely concern political and technological terms and all types of LSC consequences denoted by [Ullmann \(1951, 1962\)](#) will be observable" seems to have been correct. However, as mentioned in subsections 4.4 and 4.5, some of the detected changes are actually the result of annotation errors and the influence of context, which is why the results of this study must be taken cautiously.

5 Ethical Considerations

When conducting research of any kind, ethical considerations have to be made. The most pressing issue concerning this research is its environmental impact. Each step in the methodology required different amounts of computational power. Especially computationally heavy was the process of running XL-Lexeme, for which GPU was utilised. What is more, each piece of code was run multiple times due to numerous corrections and methodology changes that had to be made in the process. The significant amount of computational power used in this project led to increased energy consumption, which directly influences the emissions of CO_2 and other greenhouse gases. Although a certain level of environmental impact was unpreventable, efforts were made to ensure that the code was energy-efficient and that the number of code trials was kept at the minimal required level.

Another issue is that of consent. Although the data present in the PPC corpus comes from parliamentary meetings, which are publicly available, the politicians whose speeches were utilised in this project are not aware that their words are being used for the purpose of LSC detection. Although LSC detection would most probably not be deemed a controversial goal, there exists a chance that the speakers would opt to act differently knowing that their speeches would be used in such a project. To show respect and lack of prejudice towards the speakers, the author does not comment on or judge the content of the speeches, excerpts of which are presented in this paper. To ensure additional anonymity of the speakers, the names of the authors of the quoted passages are not disclosed.

6 Critiques and Limitations

Although the research on LSC detection in the Polish language can be considered quite innovative for the field, the results obtained in this project must be considered together with its limitations to ensure an objective and informed perspective.

Firstly, the use of the Polish Parliamentary Corpus can be seen as both a strength and a limitation of this study. The topics discussed in the parliament often coincide with major historical events that oftentimes influence LSC. While this can be seen as the strength of the corpus, it also implies that some of the words connected to more down-to-earth, mundane topics are not examined. The general style of the speeches and the way politicians use certain words can be seen in the example sentence below:

[...] świadomość, że w mieście dziś rządzi klasa robotnicza, która w zarodku dławi spekulantów, wyzyskiwaczy jako wrogów klasy robotniczej i warstwy chłopskiej, wrogów całej demokracji ludowej.

‘[...] the awareness that in the city today the working class rules, nipping in the bud the speculators, the exploiters as enemies of the working class and the peasant layer, enemies of the whole of popular democracy.’

This limited scope of topics covered by the parliamentary speeches gives rise to results with restricted generalisability. To frame it differently, changes to the words found in everyday speech may have remained undiscovered due to the legislative context of the PPC.

Another issue connected to the use of the PPC is the significant influence of context on LSC detection, as shown in subsection 4.5. This influence of context substantially limits the reliability of the results, putting into doubt whether some of the changes detected in this study have actually occurred. While the influence of context is a perpetual issue in studies of this kind, there is a possibility that a different corpus would yield more reliable results. Nevertheless, it is virtually impossible to extract the “general truths” from any corpus. The most one could find from corpus analysis is the trends occurring in the studied dataset, which can never be generalised to the whole language with full confidence, given the complexity of human languages. Moreover, at this point, it is worth noting that publicly available resources for linguistic analysis of the Polish language are rather scarce, essentially amounting to works of literature and a limited amount of press releases. The PPC was therefore the only corpus of transcribed spontaneous speech that could, to some extent, reflect naturally occurring language.

Furthermore, the size of the PPC, although substantial, may have been too small to yield more reliable results. Compared to the corpora used for the SemEval2020 task 1, some of which reach 71.0M tokens in t_1 and 110.0M tokens in t_2 (Schlechtweg et al., 2020), the PPC falls significantly behind in terms of scale. At the same time, however, the PPC is the only corpus of Polish transcribed speech that was publicly available at the time of conducting this study. This limitation highlights the need for further efforts in corpora collection for Polish spontaneous speech.

Another limitation of the current study is the previously mentioned errors in the annotation of the PPC. OCR lemmatisation errors, and non-expanded abbreviations all contributed to the noise in the data that significantly influenced the results. If it were not for the manual checking of the results and their careful analysis, there would be numerous errors among the lists of the most changed words. This issue stems from the imperfect nature of computational tools used for annotating large textual corpora. Such a problem is especially likely to occur when annotating texts in understudied languages such as Polish, which, once again, underlines the need to develop tools for analysing languages like Polish.

The final issue lies within the code that extracts sentences from the annotated files (subsection 3.2.5). Ideally, such an extraction would be performed with an XML parser and conditional loops. However, due to the

limited computational resources available, regular expressions were utilised instead. Regular expressions leveraged for this purpose tended to cut sentences short whenever a period (‘.’) was spotted. Periods, however, did not always indicate the end of the sentence—they could also imply an abbreviation, which was often the case, as in the example below:

A jeden z leaderów obecnie rządzącej większości, p.

‘And one of the leaders of the current ruling majority, Mr.’

For this reason, some word embeddings were created out of unfinished sentences, which might have potentially impacted the final cosine distance scores for certain lemmas, although the exact type of impact is not certain. If the project was to be improved, more refined regular expressions would have to be created.

7 Conclusion

This study covered the topic of automatic LSC detection in the Polish language. It is an innovative study with regard to its linguistic scope, as the Polish language is largely understudied in the field of NLP and no previous records of Polish LSC detection are available at the point of writing this thesis. To broaden this field's linguistic spectrum, this study utilised a bi-encoder transformer-based model, XL-Lexeme, which has a track record of state-of-the-art performance in languages such as English, Swedish, German and Russian. Using the Polish Parliamentary Corpus, two time periods were distinguished. t_1 covered the years 1919-1961 (pre-soviet and soviet periods) and t_2 spanned the years 1989-2023 (post-soviet period). Two research questions were asked at the beginning of this study: (1) how does XL-Lexeme perform with Polish data?, and (2) what kinds of change can be observed between the two predefined time periods?

The results of the study showed a robust performance of XL-Lexeme on Polish data, credibly identifying the changed words and ranking them in order of their level of change. To further confirm XL-Lexeme's reliable performance, a WiC task was performed on a small portion of annotated data (given that WiC solving ability often translates to the LSC detection ability). The results of this task suggest that the model's identification of difference in meaning is almost always correct, however it often fails to recognise differences. This indicates room for improvement in the model.

Changes detected by the model often pertain to technological and political advancements. When it comes to their types, those include semantic widening and narrowing as well as change in the meaning distribution and rare amelioration and pejoration of meaning. It is, however, worth noting that many changes were detected due to noise in the dataset and the influence of context. This urges the reader to examine the results of this study with caution and presses for further development of high-quality corpora consisting of casual spoken Polish transcriptions.

The results of this study serve as a contribution to the fields of NLP and LSC, not only by providing data on the Polish language but also by testing transformer-based contextualised embeddings on a new language. The static vs. contextualised embeddings debate may draw arguments from this study that may further aid in establishing the usefulness of both types of word representations.

8 Future Work

As already mentioned, the project described in this thesis serves as the first research on automatic LSC detection to enter the Polish language framework. This fact alone indicates the breadth of issues that could potentially be the object of future research inspired by this thesis.

This thesis employs contextual, rather than static, embeddings to perform LSC detection. This decision was influenced by the superior performance of contextual embeddings and transformer-based models on the Russian language in the RuShiftEval study (Kutuzov & Pivovarova, 2021). As Russian and Polish share many lexical and grammatical similarities, it was assumed that whatever embedding type works well for one language will also work well for the other. While this assumption seems to be valid, it cannot be said with conviction that contextual embeddings work better than static ones for the Polish language. In order to compare the embedding types and their performance in the Polish language, further research employing static embeddings is necessary. Such research would bring about a better understanding of word embeddings and their potentially changing effectiveness depending on the language type with which they are working.

Even within the domain of transformer-based models, it would be worth examining different types of models of meaning apart from XL-Lexeme. It is possible that a different model architecture or parameters could positively influence LSC detection in the Polish language. A deeper research into this topic could result in the discovery of models that could outperform XL-Lexeme.

Of course, apart from expanding the current research, future works could focus on improving this study by addressing the limitations illuminated in section 6. By doing so, more reliable and generalisable results could be obtained, supporting or questioning the conclusions drawn from this study. For example, as was mentioned in section 6, the field of NLP in the Polish language is in dire need of high-quality corpora of spontaneous speech transcriptions. If such corpora ever become available, it would be beneficial to utilise them when conducting similar research as the current one. Not only would that yield results of higher quality, but also, by comparing such research with this one, it would make it possible to state to what extent the corpus selection influences the results. Moreover, the extended availability of such corpora would enable further research into LSC in Polish across different time periods than those covered by this research.

References

- Aitchison, J. (2001). *Language Change: Progress Or Decay?* Cambridge Approaches to Linguistics. Cambridge University Press.
- Basile, P., Caputo, A., Caselli, T., Cassotti, P., & Varvara, R. (2020). *DIACR-Ita @ EVALITA2020: Overview of the EVALITA2020 Diachronic Lexical Semantics (DIACR-Ita) Task*, (pp. 411–419).
- Beaumont, N. (2021). The impact of technology on the words we use. Last accessed 7 May 2024.
- Bizzoni, Y., Degaetano-Ortlieb, S., Fankhauser, P., & Teich, E. (2020). Linguistic variation and change in 250 years of english scientific writing: A data-driven approach. *Frontiers in Artificial Intelligence*, 3.
- Blank, A. (1999). Why do new meanings occur? a cognitive typology of the motivations for lexical semantic change. In A. Blank & P. Koch (Eds.), *Historical Semantics and Cognition* (pp. 61–90). Berlin, New York: De Gruyter.
- Bloomfield, L. (1984). *Language*. University of Chicago Press.
- Britannica, E. (2024). Polish language. <https://www.britannica.com/topic/Polish-language>. Last accessed 9 May 2024.
- Cassotti, P., Caputo, A., Polignano, M., & Basile, P. (2020). GM-CTSC at SemEval-2020 task 1: Gaussian mixtures cross temporal similarity clustering. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 74–80). Barcelona (online): International Committee for Computational Linguistics.
- Cassotti, P., Siciliani, L., DeGemmis, M., Semeraro, G., & Basile, P. (2023). XL-LEXEME: WiC pre-trained model for cross-lingual LEXical sEMantic change. In A. Rogers, J. Boyd-Graber, & N. Okazaki (Eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)* (pp. 1577–1585). Toronto, Canada: Association for Computational Linguistics.
- CLARIN-PL, w. (2023). Polish parliamentary corpus. <https://clarin-pl.eu/index.php/en/kdp-en/>. Last accessed 15 May 2024.
- Conneau, A., Khandelwal, K., Goyal, N., Chaudhary, V., Wenzek, G., Guzmán, F., Grave, E., Ott, M., Zettlemoyer, L., & Stoyanov, V. (2020). Unsupervised cross-lingual representation learning at scale. In D. Jurafsky, J. Chai, N. Schluter, & J. Tetreault (Eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics* (pp. 8440–8451).: Association for Computational Linguistics.
- De Bussler, R. (2015). *Language Structure and Environment: Social, Cultural, and Natural Factors*, chapter 1, (pp. 1–28). John Benjamins.
- Duignan, B. (2024). Great vowel shift. <https://www.britannica.com/topic/Great-Vowel-Shift>. Last accessed 7 May 2024.
- Ehrmanntraut, A., Hagen, T., Konle, L., & Jannidis, F. (2021). Type-and token-based word embeddings in the digital humanities. In *CHR* (pp. 16–38).
- Eisenstein, J. (2019). Measuring and modeling language change. In A. Sarkar & M. Strube (Eds.), *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials* (pp. 9–14). Minneapolis, Minnesota: Association for Computational Linguistics.

- Elman, J. L. (1990). Finding structure in time. *Cognitive Science*, 14, 179–211.
- Firth, J. (1957). A synopsis of linguistic theory 1930-1955. In *Studies in Linguistic Analysis*. Philological Society, Oxford. reprinted in Palmer, F. (ed. 1968) *Selected Papers of J. R. Firth*, Longman, Harlow.
- Garg, N., Schiebinger, L., Jurafsky, D., & Zou, J. (2018). Word embeddings quantify 100 years of gender and ethnic stereotypes. *Proceedings of the National Academy of Sciences*, 115(16), E3635–E3644.
- Grant, A. P. (2015). Lexical borrowing. In *The Oxford Handbook of the Word*. Oxford University Press.
- Gulordava, K. & Baroni, M. (2011). A distributional similarity approach to the detection of semantic change in the Google Books ngram corpus. In S. Pado & Y. Peirsman (Eds.), *Proceedings of the GEMS 2011 Workshop on GEometrical Models of Natural Language Semantics* (pp. 67–71). Edinburgh, UK: Association for Computational Linguistics.
- Haider, T. & Eger, S. (2019). Semantic change and emerging tropes in a large corpus of New High German poetry. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change* (pp. 216–222). Florence, Italy: Association for Computational Linguistics.
- Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. In K. Erk & N. A. Smith (Eds.), *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1489–1501). Berlin, Germany: Association for Computational Linguistics.
- Harris, Z. S. (1954). Distributional structure. *WORD*, 10(2-3), 146–162.
- Hilpert, M. & Gries, S. (2016). *Quantitative approaches to diachronic corpus linguistics*, (pp. 36–53).
- Jurgens, D. & Stevens, K. (2009). Event detection in blogs using temporal random indexing. (pp. 9–16).
- Karnysheva, A. & Schwarz, P. (2020). TUE at SemEval-2020 task 1: Detecting semantic change by clustering contextual word embeddings. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 232–238). Barcelona (online): International Committee for Computational Linguistics.
- Kim, Y., Chiu, Y.-I., Hanaki, K., Hegde, D., & Petrov, S. (2014). Temporal analysis of language through neural language models. In C. Danescu-Niculescu-Mizil, J. Eisenstein, K. McKeown, & N. A. Smith (Eds.), *Proceedings of the ACL 2014 Workshop on Language Technologies and Computational Social Science* (pp. 61–65). Baltimore, MD, USA: Association for Computational Linguistics.
- Knara, I. (2017). Zapożyczenia w języku polskim. <https://blog.e-polish.eu/zapozyczenia-w-jezyku-polskim/>. Last accessed 9 May 2024.
- Kurtyigit, S., Park, M., Schlechtweg, D., Kuhn, J., & Schulte im Walde, S. (2021). Lexical semantic change discovery. In C. Zong, F. Xia, W. Li, & R. Navigli (Eds.), *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 6985–6998).: Association for Computational Linguistics.
- Kutuzov, A. & Giulianelli, M. (2020). UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 126–134). Barcelona (online): International Committee for Computational Linguistics.

- Kutuzov, A., Øvrelid, L., Szymanski, T., & Velldal, E. (2018). Diachronic word embeddings and semantic shifts: a survey. In E. M. Bender, L. Derczynski, & P. Isabelle (Eds.), *Proceedings of the 27th International Conference on Computational Linguistics* (pp. 1384–1397). Santa Fe, New Mexico, USA: Association for Computational Linguistics.
- Kutuzov, A. & Pivovarova, L. (2021). Rushifteval: A shared task on semantic shift detection for russian. In *Computational linguistics and intellectual technologies*, number 20 in Komp'uternaâ lingvistika i intellektual'nye tehnologii - Computational Linguistics and Intellectual Technologies Russian Federation: Redkollegija sbornika. International Conference on Computational Linguistics and Intellectual Technologies : Dialogue 2021 ; Conference date: 16-06-2021 Through 19-06-2021.
- Labov, W. (1997). *The Social Stratification of (r) in New York City Department Stores*, (pp. 168–178). Macmillan Education UK: London.
- Laicher, S., Kurtyigit, S., Schlechtweg, D., Kuhn, J., & Schulte im Walde, S. (2021). Explaining and improving BERT performance on lexical semantic change detection. In I.-T. Sorodoc, M. Sushil, E. Takmaz, & E. Agirre (Eds.), *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Student Research Workshop* (pp. 192–202).: Association for Computational Linguistics.
- Laine, T. & Watson, G. (2014). Linguistic sexism in The Times-A diachronic study. *International Journal of English Linguistics*, 4(3), 1.
- Lenci, A. & Sahlgren, M. (2023). *From Usage to Meaning: The Foundations of Distributional Semantics*, (pp. 3–25). Studies in Natural Language Processing. Cambridge University Press.
- Li, Y., Breithaupt, F., Hills, T., Lin, Z., Chen, Y., Siew, C. S. Q., & Hertwig, R. (2024). How cognitive selection affects language change. *Proceedings of the National Academy of Sciences*, 121(1), 1–20.
- Liu, Q., Ponti, E. M., McCarthy, D., Vulić, I., & Korhonen, A. (2021). AM2iCo: Evaluating word meaning in context across low-resource languages with adversarial examples. In M.-F. Moens, X. Huang, L. Specia, & S. W.-t. Yih (Eds.), *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing* (pp. 7151–7162). Online and Punta Cana, Dominican Republic: Association for Computational Linguistics.
- Mantiri, O. (2010). Factors affecting language change. *SSRN*, (pp. 1–11).
- Mei, T. (2020). From static embedding to contextualized embedding. <https://ted-mei.medium.com/from-static-embedding-to-contextualized-embedding-fe604886b2bc>. Last accessed 9 May 2024.
- Mitra, S., Mitra, R., Riedl, M., Biemann, C., Mukherjee, A., & Goyal, P. (2014). That's sick dude!: Automatic identification of word sense change across different timescales. In K. Toutanova & H. Wu (Eds.), *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 1020–1029). Baltimore, Maryland: Association for Computational Linguistics.
- Mudadla, S. (2023). Bi-encoder vs cross encoder? when to use which one? <https://medium.com/@sujathamudadla1213/bi-encoder-vs-cross-encoder-when-to-use-which-one-4a20edbe6d37>. Last accessed 9 May 2024.
- Mufwene, S. S. (2001). *The Ecology of Language Evolution*. Cambridge Approaches to Language Contact. Cambridge University Press.
- Nevalainen, T., Säily, T., & Vartiainen, T. (2020). Comparative sociolinguistic perspectives on the rate of linguistic change. *Journal of Historical Sociolinguistics*, 6(2).

- Nordquist, R. (2019). Language change. <https://www.thoughtco.com/what-is-a-language-change-1691096>. Last accessed 7 May 2024.
- Ogrodniczuk, M. (2018). Polish Parliamentary Corpus. In D. Fišer, M. Eskevich, & F. de Jong (Eds.), *Proceedings of the LREC 2018 Workshop ParlaCLARIN: Creating and Using Parliamentary Corpora* (pp. 15–19). Paris, France: European Language Resources Association (ELRA).
- Okrent, A. (2019). 13 words that changed from negative to positive meanings (or vice versa). <https://www.mentalfloss.com/article/65987/13-words-changed-negative-positive-or-vice-versa>. Last accessed 7 May 2024.
- Periti, F. & Tahmasebi, N. (2024). A systematic comparison of contextualized word embeddings for lexical semantic change.
- Perrone, V., Hengchen, S., Palma, M., Vatri, A., Smith, J. Q., & McGillivray, B. (2021). Lexical semantic change for ancient greek and latin. *Computational approaches to semantic change*, (pp. 287–310).
- Podlaski, M. (2017). Stare wyrazy, nowe znaczenia. o neosemantyzmach w języku. <https://zpe.gov.pl/b/stare-wyrazy-nowe-znaczenia-o-neosemantyzmach-w-jezyku/PhXHdRR93>. Last accessed 7 May 2024.
- Poudel, S. (2023). Recurrent neural network (rnn) architecture explained. <https://medium.com/@poudelsushmita878/recurrent-neural-network-rnn-architecture-explained-1d69560541ef>. Last accessed 9 May 2024.
- Pranshu, S. (2024). Introduction to feed-forward neural network in deep learning. <https://www.analyticsvidhya.com/blog/2022/03/basic-introduction-to-feed-forward-network-in-deep-learning/>. Last accessed 9 May 2024.
- Rodrigues, F. & Ravasco Nobre, C. (2020). Language/culture: Two inseparable faces of a second language teaching. In *12th International Conference on Education and New Learning Technologies* (pp. 2219–2226).
- Sagi, E., Kaufmann, S., & Clark, B. (2009). Semantic density analysis: Comparing word meaning across time and phonetic space. In R. Basili & M. Pennacchiotti (Eds.), *Proceedings of the Workshop on Geometrical Models of Natural Language Semantics* (pp. 104–111). Athens, Greece: Association for Computational Linguistics.
- Saleem, S. (2023). Neural networks in 10 mins. simply explained! <https://medium.com/@sadafsaleem5815/neural-networks-in-10mins-simply-explained-9ec2ad9ea815>. Last accessed 12 May 2024.
- Schlechtweg, D., McGillivray, B., Hengchen, S., Dubossarsky, H., & Tahmasebi, N. (2020). SemEval-2020 task 1: Unsupervised lexical semantic change detection. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 1–23). Barcelona (online): International Committee for Computational Linguistics.
- Starmer, J. (2023). Transformer neural networks, chatgpt's foundation, clearly explained!!! <https://www.youtube.com/watch?v=zxQyTK8quyY&t=1353s>.

- Tahmasebi, N., Borin, L., & Jatowt, A. (2021). *Computational approaches to semantic change*, chapter 1. Number 6 in Language Variation. Language Science Press: Berlin.
- Tahmasebi, N., Borin, L., Jatowt, A., & Xu, Y., Eds. (2019). *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, Florence, Italy. Association for Computational Linguistics.
- Tahmasebi, N. & Dubossarsky, H. (2023). Computational modeling of semantic change.
- Ullmann, S. (1951). *The Principles of Semantics*. Glasgow University publications. Jackson.
- Ullmann, S. (1962). *Semantics: An Introduction to the Science of Meaning*. A Blackwell paperback. Basil Blackwell.
- Urban, M. (2015). Lexical semantic change and semantic reconstruction. In *The Routledge handbook of historical linguistics* (pp. 374–392). Routledge.
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. u., & Polosukhin, I. (2017). Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, & R. Garnett (Eds.), *Advances in Neural Information Processing Systems*, volume 30: Curran Associates, Inc.
- Vylomova, E., Murphy, S., & Haslam, N. (2019). Evaluation of semantic change of harm-related concepts in psychology. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change* (pp. 29–34). Florence, Italy: Association for Computational Linguistics.
- Wevers, M. (2019). Using word embeddings to examine gender bias in Dutch newspapers, 1950-1990. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change* (pp. 92–97). Florence, Italy: Association for Computational Linguistics.
- Zamora-Reina, F. D. & Bravo-Marquez, F. (2020). DCC-uchile at SemEval-2020 task 1: Temporal referencing word embeddings. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 194–200). Barcelona (online): International Committee for Computational Linguistics.
- Zhou, J. & Li, J. (2020). TemporalTeller at SemEval-2020 task 1: Unsupervised lexical semantic change detection with temporal referencing. In A. Herbelot, X. Zhu, A. Palmer, N. Schneider, J. May, & E. Shutova (Eds.), *Proceedings of the Fourteenth Workshop on Semantic Evaluation* (pp. 222–231). Barcelona (online): International Committee for Computational Linguistics.

A Resources

A.1 PRT Top 100

The words that were omitted during analysis due to being proper names or having OCR or lemmatisation errors are marked in bold. The numbers are cosine distances.

1. 'oświetlić' - 0.5685421634821135
2. 'dowodzenie' - 0.43811566998367
3. 'wszczepiać' - 0.42638901388940076
4. 'wypływ' - 0.42471628447596654
5. **'jon'** - **0.4196669853663104**
6. 'wszczepić' - 0.3890992421807298
7. **'porazić'** - **0.3851678254458135**
8. 'oświetlać' - 0.3841718557446461
9. 'korona' - 0.3828625191202494
10. **'CD'** - **0.3798152635873183**
11. **'zubożać'** - **0.37452787379912234**
12. 'marka' - 0.37152475670705276
13. **'trzeć'** - **0.36226846745494135**
14. 'obchodzić' - 0.35928573904082084
15. **'Lida'** - **0.356059815045664**
16. 'służąca' - 0.3544516460728203
17. 'zadany' - 0.3517808561010327
18. 'aktyw' - 0.3449572332288613
19. 'markowy' - 0.3420346383281726
20. 'prąd' - 0.3404845942775532
21. 'przeszczepić' - 0.33365265764972096
22. **'gen'** - **0.3334135080991677**
23. 'obróbka' - 0.32906995330709365
24. 'krótkofalowy' - 0.32568188075438564
25. 'zarodek' - 0.3222359411642802
26. **'leader'** - **0.3216409557450026**
27. 'przeszczepiać' - 0.32094827003142457
28. 'wnioskować' - 0.3164608717710993
29. **'kod'** - **0.3132220506594833**
30. 'oświetlenie' - 0.3114842665840004
31. 'zmierzyć' - 0.30911513652090417
32. **'klasy'** - **0.30843301418111413**
33. 'zapracowany' - 0.3069412569390131
34. **'koda'** - **0.30689487475310884**
35. 'sok' - 0.30628175588075623
36. **'pet'** - **0.30283616597572693**
37. **'oddal'** - **0.29987277929017886**
38. 'skrzynia' - 0.2987001679974415
39. **'Lot'** - **0.2976367162304969**
40. **'minia'** - **0.2975390104384479**
41. 'zatrzymywać' - 0.29426211755011333
42. 'przywołać' - 0.29320400396156754
43. 'cyfrowy' - 0.29312565521686973
44. 'doktor' - 0.2914940381721338
45. **'woźny'** - **0.28914167961632553**
46. 'horyzont' - 0.28257138570961426
47. 'poczytać' - 0.2823349596479021
48. 'rozstrój' - 0.28219905196649575
49. 'generacja' - 0.2815660816875115
50. 'błogosławiony' - 0.28141297447470015
51. 'zgwalcić' - 0.28020954757411576
52. 'wygładzać' - 0.2777094679959963
53. 'obywatelka' - 0.27620685631999
54. 'zaklinać' - 0.27586422052483606
55. 'telegraficzny' - 0.27484360311792067
56. **'poić'** - **0.27370685292440466**
57. 'wegetacyjny' - 0.272647178372436
58. 'niekwalifikowany' - 0.2669793339731704
59. 'nieczytelny' - 0.2663741362767319
60. **'Rado'** - **0.2596441437679011**
61. **'uprawić'** - **0.25764720559119214**
62. **'osad'** - **0.2554959718488182**
63. 'naznaczyć' - 0.2553341516290458
64. **'ki'** - **0.25358887032555577**
65. 'studium' - 0.25194420094300574
66. **'mililitr'** - **0.2503026302718412**
67. 'rozjeżdżać' - 0.24995485003864337
68. **'Ita'** - **0.24934916252588035**
69. 'budynkowy' - 0.2487450859375584
70. 'oddalić' - 0.24797141235750353

71. 'rozkładowy' - 0.2458315494537221
72. 'metropolia' - 0.24481645046328626
73. 'treściwy' - 0.24365687889651766
74. 'korespondencyjny' - 0.24313891679254307
75. 'socjalizacja' - 0.242545052857565
76. 'odstępować' - 0.24155197159514785
77. 'wiekowy' - 0.24013670893539496
78. 'tykać' - 0.24005774836290839
79. 'syntetyczny' - 0.23998031925915753
80. 'reparacja' - 0.23960152460562634
81. 'przekładać' - 0.23958070009835852
82. **'ii' - 0.23606053556248352**
83. **'służący' - 0.23531584113398873**
84. **'mucha' - 0.23372249661216526**
85. 'prac' - 0.23342892351813882
86. **'Głos' - 0.23342555672672083**
87. **'dera' - 0.23311650940449802**
88. 'opiewać' - 0.23307953868218767
89. 'opancerzyć' - 0.22814651959663657
90. 'sarna' - 0.2271277775198458
91. **'piąc' - 0.22685280537183528**
92. 'naczynie' - 0.22643058457475906
93. **'drenować' - 0.22543219881553134**
94. 'kreować' - 0.22509917298655335
95. 'zaabsorbować' - 0.22287744944320564
96. 'łożysko' - 0.22203180272032208
97. 'niedołączny' - 0.22142858875660154
98. 'ustawiczny' - 0.22060230539700798
99. 'ważnie' - 0.21988660466227372
100. 'rewolucyjny' - 0.2187586325417492

A.2 APD Top 100

The words that were omitted during analysis due to being proper names or having OCR or lemmatisation errors are marked in bold. The numbers are cosine distances.

1. 'oświetlić' - 0.813405767083168
2. **'CD'** - 0.8048126250505447
3. **'bajt'** - 0.801545649766922
4. 'marka' - 0.7926871627569199
5. 'ubiec' - 0.788612112402916
6. **'V'** - 0.7867936491966248
7. 'obchodzić' - 0.7830693870782852
8. **'porazić'** - 0.7829201966524124
9. 'zatrzymywać' - 0.7784185111522675
10. **'ii'** - 0.777704268693924
11. **'trzeć'** - 0.7736916244029999
12. **'X'** - 0.7733278274536133
13. 'rozjeżdżać' - 0.7709076106548309
14. **'r'** - 0.7705190479755402
15. 'korona' - 0.7698668390512466
16. 'tykać' - 0.7696415036916733
17. 'przewód' - 0.7673631608486176
18. 'regeneracja' - 0.7673029750585556
19. 'stara' - 0.7656255066394806
20. 'wybić' - 0.7631848603487015
21. 'oddalić' - 0.7626069784164429
22. **'kompania'** - 0.761567622423172
23. **'koda'** - 0.7599683851003647
24. 'przejście' - 0.7598349153995514
25. **'Le'** - 0.759299710392952
26. 'wszczepiać' - 0.7587474882602692
27. 'czysty' - 0.7586304694414139
28. 'zmierzyć' - 0.7585367858409882
29. **'k'** - 0.7579182684421539
30. 'przełożyć' - 0.757821723818779
31. **'piąć'** - 0.757178857922554
32. 'dowodzenie' - 0.755545511841774
33. 'oświetlać' - 0.7547511905431747
34. **'Undy'** - 0.7545529007911682
35. 'zdać' - 0.7539800554513931
36. 'markowy' - 0.7539612054824829
37. 'imać' - 0.7537922412157059
38. 'skrzynia' - 0.7531150877475739
39. **'minia'** - 0.7529056817293167
40. 'wypływ' - 0.7522319108247757
41. 'zarodek' - 0.7516147941350937
42. **'R'** - 0.7515168935060501
43. 'krótkofalowy' - 0.7512838840484619
44. 'zdjęcie' - 0.7509817630052567
45. **'pet'** - 0.7506987750530243
46. 'wylot' - 0.7499591410160065
47. 'oświetlenie' - 0.7499370574951172
48. **'sadzić'** - 0.7490958571434021
49. **'gram'** - 0.748613178730011
50. 'zadany' - 0.7483164966106415
51. **'służąca'** - 0.748225063085556
52. 'sok' - 0.7481521368026733
53. 'poić' - 0.748018741607666
54. 'oddal' - 0.7479535639286041
55. 'zjechać' - 0.7479184567928314
56. *'militr'* - 0.7472060918807983
57. 'wybijac' - 0.747021496295929
58. 'ciągnąć' - 0.7469871342182159
59. 'ważyć' - 0.7465983927249908
60. **'C'** - 0.7458766996860504
61. **'VI'** - 0.745346188545227
62. 'przebrać' - 0.7448324263095856
63. 'wyprowadzać' - 0.7443479001522064
64. 'zajmujący' - 0.7441484034061432
65. 'wyskakiwać' - 0.7441477179527283
66. 'doić' - 0.7440444827079773
67. **'III'** - 0.7440338730812073
68. 'zubożać' - 0.7440122663974762
69. **'Lida'** - 0.7439725697040558
70. 'prąd' - 0.7434786260128021
71. 'wydostać' - 0.7431654334068298
72. 'mijać' - 0.7426692247390747
73. 'zagrać' - 0.7419054806232452
74. **'klasy'** - 0.7414854764938354

75. 'powieść' - 0.7414206862449646
76. **'iii' - 0.741320937871933**
77. 'wzruszać' - 0.7409895360469818
78. 'para' - 0.7409291863441467
79. **'uprawić' - 0.7409277558326721**
80. **'jon' - 0.7407156229019165**
81. 'wzruszyć' - 0.7403157949447632
82. 'wykręcać' - 0.7382981181144714
83. 'studium' - 0.7380312979221344
84. **'VIII' - 0.7380189597606659**
85. 'zatrzymać' - 0.7376811504364014
86. 'wnioskować' - 0.7369494736194611
87. 'przewrócić' - 0.7363288700580597
88. 'udawać' - 0.7354360520839691
89. 'zaabsorbować' - 0.7354265749454498
90. 'przywołać' - 0.7353865802288055
91. 'przebierać' - 0.7350754141807556
92. 'dobijać' - 0.73504838347435
93. 'przebić' - 0.7347200214862823
94. 'doktor' - 0.7337708473205566
95. 'rozkładowy' - 0.7336130142211914
96. **'kora' - 0.7335449457168579**
97. 'odstępować' - 0.7334012687206268
98. 'zapracowany' - 0.7327242195606232
99. 'żelazny' - 0.7324277758598328
100. **'M' - 0.7318454682826996**