# Can Large Language Models compete with specialized models in Lexical Semantic Change Detection?

**Frank D. Zamora-Reina**[a,*], **Felipe Bravo-Marquez**[a], **Dominik Schlechtweg**[b] **and Nikolay Arefyev**[c]

[a]Department of Computer Science, CENIA & IMFD, University of Chile
[b]Institute for Natural Language Processing, University of Stuttgart, Stuttgart, Germany
[c]Department of Informatics, University of Oslo, Oslo, Norway

**Abstract.** In this paper, we present a comprehensive comparison between specialized Lexical Semantic Change Detection (LSCD) models and Large Language Models (LLMs) for the LSCD task. In addition to comparing models, we also investigate the role of automatic prompt selection for improving LLM performance. We evaluate three approaches: Average Pairwise Distance (APD), Word-in-Context (WiC), and Word Sense Induction (WSI). Using Spearman correlation as the evaluation metric, we assess the performance of Mixtral, Llama 3.1, Llama 3.3, and specialized LSCD models across English and Spanish datasets. Our results show that by using prompt optimization and LLMs, we achieve state-of-the-art performance for the English dataset and outperform specialized LSCD models at the annotation level in the same dataset. For Spanish, specialized models outperform LLMs across all three approaches—WiC, APD, and WSI—indicating that specialized LSCD models are still more effective for semantic change detection in Spanish.

## 1 Introduction

Lexical Semantic Change Detection (LSCD) is a task that investigates the evolution of word meanings over time through two primary subtasks: (i) binary classification, which aims to determine whether a word has changed its meaning due to the acquisition or loss of senses, and (ii) the ranking task, which seeks to assess the degree of Lexical Semantic Change (LSC) for a set of target words by comparing their sense frequency distributions in two different time periods [33, 42]. In this research, we focus exclusively on the ranking task.

Over time, LSCD has benefited from the development of progressively more advanced methods to capture meaning variation. Early approaches relied on counting-based techniques and static word embeddings [16, 38, 37], but these were soon surpassed by contextualized models [25], which enabled more nuanced representations of word usage in context. Among these, models trained on the Word-in-Context (WiC) task have proven particularly effective for LSCD, since both tasks involve determining whether a pair of word usages share the same meaning. The key difference lies in LSCD's diachronic nature, where usage comparisons span across different time periods.

Notably, DeepMistake [2] and XL-LEXEME [4] have demonstrated state-of-the-art performance on several LSCD datasets. DeepMistake achieved leading results in the Russian [15] and Spanish [42] shared tasks, while XL-LEXEME outperformed previous methods on English, Swedish, and German datasets. These models leverage the close relationship between the LSCD and WiC tasks to learn effective contextualized representations for semantic change.

More recently, Large Language Models (LLMs) such as GPT-3 [3] have emerged as general-purpose tools capable of strong performance across a wide variety of Natural Language Processing (NLP) tasks [43, 21]. These models are trained on massive text corpora and can generate high-quality responses from natural language prompts alone, without task-specific fine-tuning. Despite their success, their applicability to LSCD—especially for the ranking task—has not been applied widely.

Despite the widespread success of LLMs across various NLP tasks, their performance has been shown to be highly sensitive to the choice of prompt [17, 45]. This sensitivity highlights a key limitation: crafting effective prompts often requires substantial human expertise and manual effort. Previous studies have shown that optimized prompts can significantly enhance model performance compared to unrefined or manually designed prompts, yielding improvements in tasks such as classification, question answering, and others [44, 7, 6].

In this paper, we aim to compare specialized LSCD models with general-purpose LLMs for the LSCD task. This comparison is motivated by the practicality of LLMs: (i) they can be applied without fine-tuning on downstream tasks, and (ii) given the promising outcomes reported across several researches [43, 21]. In addition to the model comparison, we also explore whether automatic prompt selection can enhance the semantic proximity judgments produced by LLMs and check if those models can outperform much smaller and faster specialized models.

To achieve our goals, we propose to answer the following research questions:

- **RQ1.** Can automatically optimized prompts yield better results for the LSCD task than manually crafted prompts designed through prompt engineering?
- **RQ2**. Can LLMs solve the Graded Change LSCD task well? Can these results surpass the WiC models reported as state-of-the-art?
- **RQ3**. Can LLMs outperform state-of-the-art LSCD models at the annotation level?

## 2 Related work

Graded Change Ranking LSCD is defined as follows [33, 42]:

* Corresponding Author. Email: fzamora.reina@gmail.com

Given a set of target words and a set of uses $U_1$ and $U_2$ for each of them, the task is to rank the target word according to their degree of Lexical Semantic Change between $U_1$ and $U_2$.

The gold graded change scores of each word is computed as the Jensen-Shannon distance (JSD) of the sense frequency distributions for two time periods, thus generating a list of words sorted by the degree of semantic change [33, 42, 35]. Then, Spearman correlation [36] is the metric used between the gold scores and the predicted word scores.

**LLMs + LSCD** Wang and Choi [40] demonstrate that the use of LLMs, specifically GPT-4 [1], yields superior results compared to BERT [5] and traditional methods, including statistical approaches and static embedding techniques [20], for detecting semantic change. The authors employ an experimental setup based on the TempoWiC dataset [18], which involves two tasks: instance-level meaning shift detection and corpus-level lexical semantic change (LSC) detection. In the first task, they utilize a binary classification approach to determine whether a target word has changed its meaning within the entire corpus. The second task involves classifying whether a target word retains similar meanings across sentence pairs from two different time periods. Predictions from the LLMs are generated using various prompts. Ultimately, the authors demonstrate that LLMs outperform the other methods in both tasks.

**LLMs + WiC** In a similar vein, Periti et al. [27] present a series of experiments comparing the performance of the BERT model [5] with that of ChatGPT, utilizing both the web interface and the foundation model via the API. Rather than employing LSCD datasets, the authors evaluate ChatGPT's performance on the WiC task [28], given its relevance to LSCD [2, 4].

To achieve this, they utilize two datasets: TempoWiC [18] and DWUG EN [33].[1] TempoWiC is employed to evaluate changes in meaning over a short time span, while DWUG EN assesses the model's ability to detect semantic change in a long time span. Predictions from the LLMs are generated using various prompts. The results indicate that BERT outperforms the LLMs in this evaluation. Ultimately, the authors conclude that ChatGPT faces challenges in detecting semantic change, both in diachronic contexts and over short time periods. These results are contradictory to the studies of Wang and Choi [40]; however, the experimental setups in both cases differ significantly.

Yadav et al. [41] find that automating the annotation process by reusing human-tailored instructions is a major challenge. To substantiate their claims, they use the DWUG EN dataset [34] comprising 46,000 pairs of word usages annotated following a 4-point semantic relatedness scale: 1 (unrelated meanings), 2 (distantly related meanings), 3 (closely related meanings), and 4 (identical meanings) [32]. By utilizing ChatGPT through its API, the authors design a series of prompts to guide the annotation of LSCD samples. They divide the dataset into three subsets: training, development, and testing, to optimize and evaluate the prompts for achieving accurate predictions from ChatGPT. The results indicate that ChatGPT performs poorly in annotating the samples when prompted with the full annotation guidelines presented in a human-like style particularly when assessed using the Krippendorff's alpha score [13]. In contrast, ChatGPT yields better results when custom prompts are employed.

Periti and Tahmasebi [26] present a systematic comparison of contextualized models for the LSCD task. In one of their experiments, they evaluate several models used as computational annotators, including GPT-4 only for English. Their findings show that GPT-4 achieves the highest Spearman correlation with human annotations for the English dataset, outperforming all other models, including XL-LEXEME. In addition, GPT-4 also shows superior performance in the Word Sense Induction (WSI) task. However, its performance in the WiC task falls short compared to that of XL-LEXEME.

However, their work does not address the challenges or potential of prompt engineering when applying LLMs to LSCD. In particular, they do not explore how prompt formulation can impact model performance—a key consideration when working with general-purpose LLMs.

**WiC model + LSCD** The state-of-the-art models for the LSCD task consist of two WiC models. The DeepMistake model, introduced by Arefyev et al. [2], is built on the XLM-R encoder. This model consists of two components: an XLM-R-based encoder jointly encoding two word usages and a classification head. All weights are trained on the MCL-WIC dataset [19], and then undergo language-specific fine-tuning. For Spanish they are fine-tuned on the Spanish version of the synchronic XL-WSD [23], and the development version of the DWUG ES [42] datasets. For Russian the RuSemShift [30] dataset is employed. DeepMistake is the state-of-the-art for LSCD in Spanish [2] and Russian [8].

The XL-LEXEME model [4] is a pre-trained bi-encoder built on the XLM-R multilingual model. This model has been trained and fine-tuned on the MCL-WiC [19] and XL-WiC [29] datasets. XL-LEXEME currently holds the state-of-the-art position for English, Swedish, and German datasets [33].

Our work provides a systematic comparison between general-purpose, open-weight LLMs and specialized LSCD models for the LSCD task. Motivated by the practical advantages of LLMs—such as avoiding the need for task-specific fine-tuning—we assess their performance across multiple evaluation methodologies and datasets. Unlike previous studies, our work introduces the use of automatic prompt optimization as a key strategy to enhance LLM performance for LSCD, aiming to adapt these models more effectively to the task. In doing so, we prioritize a comprehensive evaluation of their capabilities relative to dedicated LSCD models.

## 3 Data

In this research, we use two datasets: DWUG EN[2] [33] and DWUG ES[3] [42]. Both datasets contain two corpora, representing text from an old period of time and a modern period of time (see Table 1). The methodology used to annotate both datasets follows the 4-point scale proposed in Schlechtweg et al. [32], where (1) represents sentence pairs where the target word has unrelated meanings, and (4) represents identical meanings.

**Table 1.** Overview of datasets. LGS = languages, Id = identifier of the dataset, n = number of target words, $C1$ = old period of time, $C2$ = modern period of time.

| LGS | Id | n | C1 | C2 |
|---|---|---|---|---|
| Spanish | DWUG ES | 60 | 1810-1906 | 1994-2020 |
| English | DWUG EN | 37 | 1810-1860 | 1960-2010 |

Additionally, we employ the DWUG EN resampled dataset[4], along with the development version of the DWUG ES dataset (the test set is

[1] Periti et al. [27] renamed this dataset as HistoWiC.

[2] https://zenodo.org/records/7387261

[3] https://zenodo.org/records/6433667

[4] https://zenodo.org/records/14025941

reported in Table 1). As will be detailed later, these two datasets are utilized to optimize the prompts designed for the LLMs to determine whether a pair of word usages have the same meaning. For this optimization, we partition the resampled DWUG EN and the development version of the DWUG ES datasets into training, development, and test sets, allocating 60% of the data for training and 20% each for development and testing.

Table 2 presents the frequency distribution of annotation labels in both datasets, which are used to optimize two prompts: one in English and the other in Spanish. As observed, the classes are imbalanced. To address this issue, we apply stratified random sampling to ensure equal-sized samples for each class across the training, development, and testing sets.

**Table 2.** Class distribution across DWUG EN resampled and development version of DWUG ES datasets.

| Dataset | Class 1 | Class 2 | Class 3 | Class 4 |
|---|---|---|---|---|
| DWUG EN (resampled) | 641 | 1,658 | 1,472 | 2,075 |
| DWUG ES (dev) | 1,406 | 1,522 | 2,343 | 3,433 |

After implementing a stratified sampling strategy, we obtain balanced samples for each class across all partitions in both datasets, as shown in Table 3. This balanced allocation enhances the robustness of training, evaluation, and testing for both languages, effectively addressing the imbalances present in the original datasets.

**Table 3.** Statistics for each partition of the DWUG EN (resampled) and the development version of the DWUG ES datasets used to optimize the prompts for annotating semantic proximity examples.

| Datasets | Class 1 | Class 2 | Class 3 | Class 4 | Target words |
|---|---|---|---|---|---|
| DWUG EN (resampled) training | 300 | 300 | 300 | 300 | 15 |
| DWUG EN (resampled) dev | 100 | 100 | 100 | 100 | 15 |
| DWUG EN (resampled) test | 100 | 100 | 100 | 100 | 15 |
| DWUG ES (dev) training | 500 | 500 | 500 | 500 | 20 |
| DWUG ES (dev) dev | 200 | 200 | 200 | 200 | 20 |
| DWUG ES (dev) test | 225 | 225 | 225 | 225 | 20 |

## 4 Experimental setup

In this section, we present the proposed methodologies designed to demonstrate the feasibility of using large language models to tackle the Graded Change LSCD task.

As outlined in Section 3, our experiments utilize two datasets: DWUG EN and DWUG ES. We generate LSC predictions using a WSI baseline approach, as described by Schlechtweg et al. [35]. Building upon the prompting strategies introduced in Section 5, we include results from both initial and optimized prompt configurations for the LLMs.[5] These prompts are designed to generate semantic proximity judgments consistent with the 4-point scale used in the manual annotations [32].

In this study, we employ three open-source language models: Llama 3.1[6], Llama 3.3, and Mixtral 8x7B[7] (Mixtral:8x7B) [11]. We specifically use the 8 billion-parameter versions of Llama 3.1 (Llama3.1:8B) and the 70 billion-parameter version of Llama 3.3 (Llama3.3:70B). We selected these models to systematically assess the impact of model scale on LSCD performance. This range enables us to examine how increasing model capacity interacts with prompt optimization and whether larger models consistently offer advantages across different evaluation approaches and languages. Due to hardware constraints, we conduct our experiments using all models in their quantized forms [9].

In the following, we describe the pipeline used to compute predictions from both LLMs and WiC models:

- **WiC** We annotate gold pairs of usages in the English and Spanish datasets. The annotations produced by XL-LEXEME [4] are the cosine distances indicating whether two word usages share the same meaning. Additionally, we employ three versions of DeepMistake: MCL->es, enMCL, and MCL, as this model has been fine-tuned on various WiC datasets across multiple languages [cf. 2, 8]. The scores produced by DeepMistake are the probabilities that two word usages have the same meaning. Finally, we prompt Llama 3.1, Mixtral, and Llama 3.3 to annotate the same usage pairs by generating semantic proximity scores aligned with the 4-point scale employed in the manual annotation of the employed datasets [32].
- **Word Usage Graph (WUG)** A weighted graph is constructed for each target word, in which the nodes represent usages of the target word, and the weighted edges correspond to the annotations predicted by the WiC models or LLMs.
- **Clustering methods** We apply one of the clustering methods to the constructed WUGs to infer word senses. Next, to quantify the semantic change for each word, we calculate the JSD between the sense frequency distributions of the inferred senses for the old and new usages.

In addition, we generate LSC scores based on the Average Pairwise Distance (APD) approach [14]:

- **APD** The average pairwise distance, derived from the scores generated by the WiC models or LLMs' annotations, is returned as a measure of semantic change for a word.

### *4.1 WUG*

We select several hyperparameters for the WUG construction procedure:

- `normalize`: If set to false, the edge weights are the raw scores provided by the models. Otherwise, they are normalized using the MinMaxScaler[8] fitted on all annotations of all words.
- `fill_diagonal`: Controls whether self-loops are inserted. Self-loops signify that a pair of identical usages has been scored.
- `threshold`: Edges with weights below this threshold are removed. During grid search, this parameter takes values at the 10th to 90th percentiles of all edge weights, plus the default value 0.5.

### *4.2 Clustering methods*

We employ in our experiments three clustering methods on WUGs: Spectral clustering[9] [39], Agglomerative clustering[10] [10], and Weighted Stochastic Block Model[11] (WSBM) [24]. Spectral clustering and Agglomerative clustering methods need to know in advance

[5] https://github.com/fdzr/optimization-prompts-dspy

[6] https://ai.meta.com/blog/meta-llama-3-1/ (for more information) and available for download at https://ollama.com/library/llama3.1.

[7] https://ollama.com/library/mixtral

[8] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html

[9] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.SpectralClustering.html

[10] https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html

[11] https://graph-tool.skewed.de/static/doc/autosummary/graphtool.inference.BlockState.html

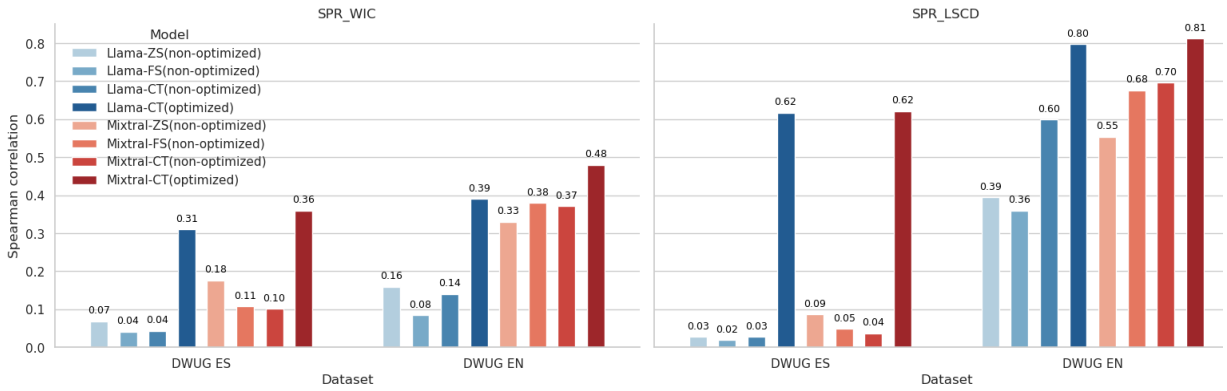


**Figure 1.** Spearman correlation between human and model annotations for word usage pairs (SPR_WiC), and between gold graded change scores and model predictions using the APD-based approach (SPR_LSCD). “ZS" denotes Zero-Shot, “FS" stands for Few-Shot, and “CT" refers to Chain-of-Thought prompting.

the number of clusters. We test values from 2 to 5 [35] and identify the optimal value using the Silhouette score [31].

The WSBM method identifies the optimal number of clusters and accommodates various distributions for drawing edge weights through the *distribution* parameter. This parameter can take the following values: `poisson`, `binomial`, `geometric`, `exponential`, and `normal`. `Poisson`, `binomial`, and `geometric` are only applicable to discrete edge values while `real` and `exponential` are only applicable to real-valued edge weights.

## 5 Prompts and their optimizations

To address **RQ1**: **Can automatically optimized prompts yield better results for the LSCD task than manually crafted prompts designed through prompt engineering?**, we design an experiment comparing the performance of LLMs using manually crafted prompts with that of LLMs using optimized prompts informed by training data.

Building on the work of Yadav et al. [41], we utilize the manually optimized prompt identified by the authors as the most effective for classifying semantic proximity samples. Originally crafted in English, this prompt was translated into Spanish to create an additional version tailored for semantic proximity samples in Spanish. Additionally, we extend the non-optimized prompts with semantic proximity examples provided to the annotators in the web of the DURel framework [32]. Based on these examples, we create three prompts:

- **Zero-shot** We provide instructions to annotate the samples using the 4-point scale, along with an explanation for each point on the scale.
- **Few-shot** We provide instructions to the model to annotate the samples using the 4-point scale, and include an example of a pair of sentences for each possible annotation.
- **Chain of Thought** This setting is the same as the few-shot approach, with the key difference that we prompt the model to reason step by step and to explain the rationale for the score it assigns to each example.

As mentioned in Section 1, hand-optimizing prompts is a challenging task, as even slight adjustments in the instructions can result in significantly different interpretations or responses from the model. To tackle this issue, we further refine both prompts using the DSPy framework [12], leveraging the data presented in Table 3.

We develop a DSPy program that employs the MIPROv2 (Multi-prompt Instruction PRoposal Optimizer) optimizer [22] in conjunction with the accuracy metric to optimize the prompts, utilizing the default parameters for this optimizer.[12] Additionally, we conduct several experiments varying the number of samples per class provided to the optimizer to explore how this parameter influences the quality of the generated prompts. We observe a consistent pattern across all settings: increasing the number of samples per class generally leads to the generation of better prompts, resulting in higher accuracy on the test set.

MIPROv2 optimizes DSPy prompts by learning how instructions and demonstrations affect task performance. It generates demonstrations via bootstrapping and proposes instructions using grounding, where a language model receives contextual information about the dataset, program, and past prompts. New configurations—combinations of instructions and demonstrations—are proposed using Bayesian optimization and evaluated on mini-batches. The best-performing configurations are then selected based on full training set performance [12].

We then annotate the test sets using both English and Spanish prompts for the English data, and both English and Spanish prompts for the Spanish data. Subsequently, we select the best prompt based on accuracy to annotate the DWUG EN and DWUG ES datasets.

**Table 4.** Accuracy of non-optimized prompts (NOP) and optimized prompts (OP) for DWUG EN and DWUG ES datasets, using prompts in English (PrE) and Spanish (PrS). Results are reported for Mixtral, Llama 3.1, and Llama 3.3 as the underlying LLMs.

| | Llama 3.1 | | Mixtral | | Llama 3.3 | |
|---|---|---|---|---|---|---|
| Dataset-Prompts | NOP | OP | NOP | OP | NOP | OP |
| DWUG ES - PrS | 26.8 | 29.5 | 32.6 | 31.22 | 32.33 | 39.77 |
| DWUG ES - PrE | 26.5 | **35.5** | 33.7 | **34.80** | 40.88 | **46.33** |
| DWUG EN - PrS | 25.75 | 31.0 | 32.8 | **40.75** | 37.25 | 45.5 |
| DWUG EN - PrE | 28.75 | **37.25** | 33.25 | 38.75 | 35.75 | **49.25** |

Table 4 presents the accuracy results for non-optimized and optimized prompts in classifying pairs of usages according to their labels on the 4-point scale. Bolded results indicate the best-performing prompt type for each dataset. Notably, the best prompt for a given dataset is not always in the same language as the dataset itself. Finally, we select the best-performing prompt for each dataset to annotate the LSCD samples.

[12] https://dspy.ai/learn/optimization/optimizers/

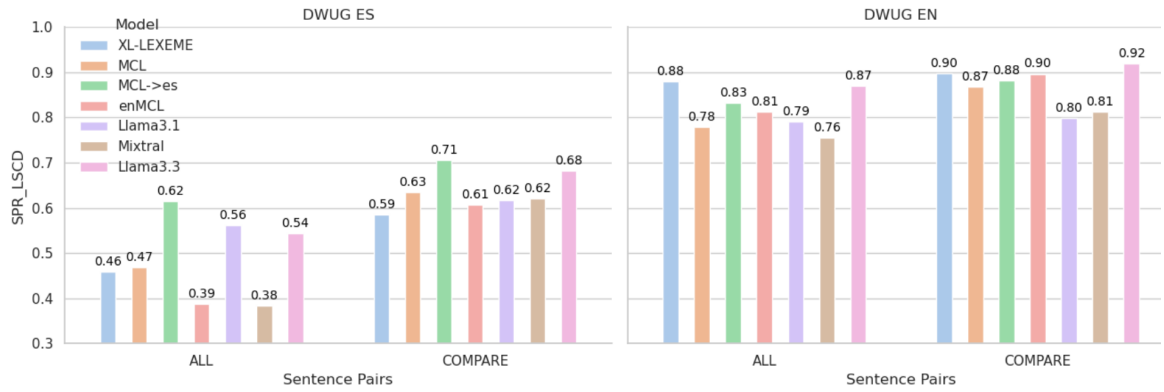


**Figure 2.** APD-based approach with various WiC models and LLMs, Spearman correlation with the gold graded change word scores is reported. "COMPARE" calculates APD across usage pairs consisting of usages from different time periods, while "ALL" employs usage pairs regardless of their time periods.

Figure 1 presents a comparison between manually engineered prompts and automatically optimized prompts, evaluated using two metrics: the Spearman correlation between human and LLM-generated annotations, and the Spearman correlation between gold graded change scores and predictions derived from the APD-based approach using LLM annotations. The results show that optimized prompts substantially outperform manually crafted ones across both languages and experimental setups.

These results show that automatic prompt optimization consistently improves the performance of medium-sized models (i.e., Llama3.1:8B and Mixtral:8x7B) on the LSCD task. Next, we extend our investigation to a larger and more recent model (specifically, Llama3.3:70B). All subsequent experiments will be conducted using LLM annotations generated with the best performing optimized prompts obtained using the methodology presented here.

## 6 Experiments for graded change detection

In this section, we describe the methodology employed to answer **RQ2** proposed earlier in Section 1.

To address this research question, we design two experiments: one based on the APD approach and the other using cross-validation. The APD-based approach does not require the selection of hyperparameters, allowing us to evaluate the entire test set. In contrast, the clustering-based approach involves several hyperparameters; therefore, we use cross-validation to select the optimal values for these hyperparameters for each of our models individually.

It is important to note that these two approaches enable us to address the Ranking LSCD task from different perspectives. APD offers a direct quantification of a word's semantic change by calculating the mean of the scores associated with usage pairs from different time periods for each word. In contrast, the cross-validation evaluation method and clustering methods focus on the underlying WSI problem.

### *6.1 APD*

We utilize the APD aggregation method to calculate the graded semantic change for each word based on annotations provided by the LLMs, DeepMistake, and XL-LEXEME. Although APD is typically applied to usage pairs originating from different time periods (COMPARE pairs), we also evaluate the performance of the APD-based approach using usage pairs from any time period (ALL pairs) to assess its effectiveness in determining the graded semantic score.

Figure 2 presents the results of this experiment. An initial observation reveals that, for English, Llama3.3:70B achieves the best performance, closely followed by XL-LEXEME and enMCL. Although Mixtral:8x7B and Llama3.1:8B yield the lowest results, their Spearman correlations of 0.81 and 0.80, respectively, remain notably high. Finally, although the performance of all models declines when using ALL sentence pairs compared to COMPARE, Llama3.3:70B and XL-LEXEME are the models least impacted by this effect.

For Spanish, the best-performing model is MCL->es, followed by Llama3.3:70B as the second-best. Mixtral:8x7B and Llama3.1:8B demonstrate similar performance—lower than MCL and Llama3.3:70B but higher than XL-LEXEME. However, when using ALL sentence pairs, the performance of all models declines significantly. Notably, in this case, Llama3.1:8B is less affected by this drop compared to Llama3.3:70B and Mixtral:8x7B.

We conclude that recent prompt optimization techniques are crucial for achieving better results on the Graded Change LSCD task, as demonstrated by the performance of Llama3.3:70B. However, medium-sized LLMs such as Mixtral:8x7B and Llama:3.1:8B still underperform compared to smaller and faster specialized LSCD models in both the DWUG EN and DWUG ES datasets. This suggests that, in addition to optimization techniques, the size of the model also significantly influences the results. Furthermore, our findings indicate that using COMPARE sentence pairs is more effective, as they more accurately capture the true relationships between sentence pairs from different time periods.

### *6.2 Cross validation with clustering*

We employ a cross-validation approach combined with hyperparameter optimization to assess the capacity of the LLMs to address the Graded Change LSCD task, as explained in Section 4. For the DWUG ES and DWUG EN datasets, we divide the sets of 60 and 37 target words into five folds. Subsequently, within each cross-validation iteration, we conduct an exhaustive hyperparameter grid search. This process encompasses hyperparameters for the WiC model, WUG, and clustering methods (see Section 4). The optimal configuration is identified in the training folds and subsequently evaluated in the testing fold. This procedure yields five performance scores (SPR_LSCD for LSCD and ARI for WSI) for each clustering

**Table 5.** Performance of various methods across cross-validation experiments on the DWUG ES and DWUG EN datasets. The table compares different clustering approaches: AC (Agglomerative Clustering), SC (Spectral Clustering), WSBM (Weighted Stochastic Block Model). APD is reported to evaluate the performance of the models in identifying the different senses of a word, as opposed to quantifying the semantic change of a word without discerning the senses of the target words. The random baseline assigns a score between 1 and 4 randomly.

| Methods | Models | Spr_LSCD (ES) | ARI (ES) | Spr_LSCD (EN) | ARI (EN) |
|---|---|---|---|---|---|
| WSBM | Llama 3.1 | .369 ± .204 | .356 ± .116 | **.845 ± .097** | .152 ± .067 |
| SC | | .271 ± .461 | .102 ± .097 | .014 ± .411 | -.03 ± .01 |
| AC | | .478 ± .286 | .073 ± .058 | .205 ± .540 | -.01 ± .03 |
| APD | | .636 ± .236 | - | .645 ± .368 | - |
| WSBM | Mixtral | .454 ± .180 | .380 ± .104 | .776 ± .219 | .161 ± .07 |
| SC | | .565 ± .141 | .092 ± .049 | -.171 ± .492 | -.03 ± .01 |
| AC | | .414 ± .075 | .068 ± .027 | -.04 ± .525 | -.003 ± .02 |
| APD | | .567 ± .332 | - | .612 ± .280 | - |
| WSBM | Llama 3.3 | .659 ± .181 | **.502 ± .09** | .729 ± .241 | .183 ± .08 |
| SC | | .507 ± .231 | .294 ± .05 | .302 ± .436 | .124 ± .113 |
| AC | | .423 ± .184 | .228 ± .05 | .195 ± .273 | -.01 ± .01 |
| APD | | .676 ± .195 | - | .752 ± .227 | - |
| WSBM | DeepMistake | **.727 ± .206** | .397 ± .074 | .730 ± .212 | .231 ± .212 |
| SC | | .561 ± .140 | .355 ± .036 | .520 ± .436 | **.273 ± .115** |
| AC | | .457 ± .320 | .341 ± .054 | .433 ± .245 | .215 ± .128 |
| APD | | .653 ± .250 | - | .638 ± .292 | - |
| WSBM | XL-LEXEME | .630 ± .377 | .452 ± .095 | .686 ± .200 | .152 ± .059 |
| SC | | .484 ± .215 | .318 ± .043 | .491 ± .176 | .137 ± .063 |
| AC | | .426 ± .255 | .292 ± .087 | .143 ± .367 | .02 ± .024 |
| APD | | .566 ± .354 | - | .814 ± .199 | - |
| WSBM | Random Baseline | -.199 ± .310 | -.02 ± .184 | -.111 ± .254 | -.05 ± .147 |

method, from which we calculate the mean and standard deviations, following the procedure proposed by Schlechtweg et al. [35]. Finally, we incorporate APD with cross-validation across all models to facilitate a comparison between the performance of models that first identify the senses of each word and a method that directly quantifies semantic change for each word.

Table 5 presents the results of the experiments conducted on the DWUG ES and DWUG EN datasets. We compare the outcomes obtained from the WSI-based approach, where predictions made by DeepMistake, XL-LEXEME, and various LLMs are used to construct WUGs.

Regarding the DWUG ES dataset, an initial observation reveals that the combination of DeepMistake and WSBM outperforms DeepMistake with APD, the latter previously considered the state-of-the-art for this dataset. Among the LLMs, Llama3.3:70B achieves the best performance, surpassing not only other LLMs but also XL-LEXEME, although it still trails behind DeepMistake combined with WSBM. Llama3.1:8B and Mixtral:8x7B perform less effectively than Llama3.3:70B, yet both still significantly outperform the random baseline.

While these findings suggest that LLMs can be valuable tools for tackling the LSCD task, they have not yet reached a level of performance that rivals the state-of-the-art model on the DWUG ES dataset. The results indicate that the annotations generated by the LLMs lack the necessary accuracy to effectively distinguish the different senses of the target words, thus limiting the interpretability of the outcomes.

The following experiment presents the results for the English benchmark. As shown in Table 5, the WSBM method combined with Llama3.1:8B emerges as the most effective approach to detect semantic change. This performance surpasses that of all other models, regardless of the clustering methods employed or APD, demonstrating Llama3.1:8B's strong ability to generalize on the DWUG EN dataset and effectively address the WSI task.

Similarly, the results obtained by Mixtral:8x7B combined with WSBM are slightly lower than those of Llama3.1:8B with WSBM, yet they surpass those of DeepMistake, XL-LEXEME, and even Llama3.3:70B across all clustering methods. This underscores Mixtral:8x7B's strong potential for the LSCD task, particularly when enhanced through techniques such as fine-tuning. Notably, both Llama3.1:8B and Mixtral:8x7B perform better with WSBM than with APD. In contrast, Llama3.3:70B combined with WSBM—its best-performing clustering method—shows a notable drop in performance compared to the APD approach, suggesting that its annotations may lack the quality needed to produce meaningful sense clusters.

The results of this study demonstrate that Llama3.1:8B is the most effective model for addressing the LSCD task using a WSI-based approach on the DWUG EN dataset. Its predictions, when combined with clustering methods, yield results that surpass those of state-of-the-art models employing similar clustering techniques, and even outperform Llama:3.3:70B, which achieved the best results using APD for English.

However, Mixtral:8x7B, Llama:3.1:8B, and Llama3.3:70B struggle to generalize effectively or produce accurate predictions on the 4-point scale for the Spanish dataset. These findings are consistent with the performance of all three models when using APD. A potential explanation for these contrasting results is the significantly larger volume of training data available for English compared to Spanish, which may have impacted the models' ability to generalize and perform accurately.

## 7 Experiments at WiC annotation level

In this section, we explore the annotations generated by all models at the annotation level. This analysis aims to answer the **RQ3** posed in Section 1: **Can LLMs outperform state-of-the-art LSCD models at the annotation level?**

The scores provided by the LLMs employed in this study correspond to annotations for all usage pairs that were also annotated by human annotators. To ensure consistency, we first filter out any usage pairs that do not have a score within the 4-point scale. We then compute the intersection between the gold-standard usage pairs and the filtered annotations from the LLMs. The remaining gold usage pair annotations are aggregated by calculating the median score across all annotators for each pair. Finally, we report the Spearman correlation between the aggregated gold annotations and the scores produced by the LLMs and WiC models.

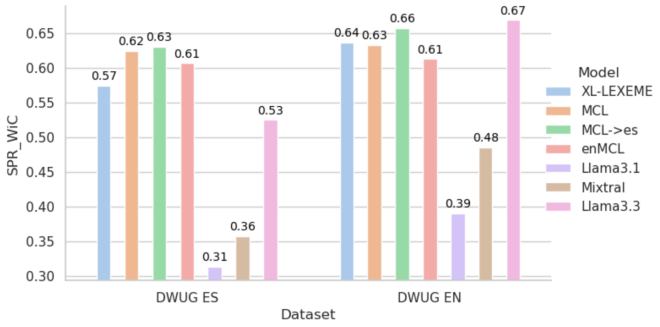


**Figure 3.** Spearman correlation between the annotations provided by human annotators, LLMs, DeepMistake models, and XL-LEXEME.

Figure 3 illustrates the Spearman correlation between the annotations provided by LLMs, DeepMistake models, and XL-LEXEME with the annotations provided by human annotators (SPR_WiC). The results across both datasets indicate that Mixtral:8x7B and Llama3.1:8B underperform compared to the other models. Notably, Llama3.3:70B achieves the highest SPR_WiC for the English dataset, which is consistent with its strong performance on the same dataset using the APD method. For Spanish, the MCL->es model achieves the highest correlation, reaffirming its effectiveness in the Spanish dataset.

The results achieved by Mixtral:8x7B and Llama3.1:8B appear somewhat contradictory: both models perform well in the graded change detection task for English, yet their performance in the annotation-level WiC task is notably low. Llama3.3:70B, on the other hand, presents the second-best performance for Spanish when using APD, but its performance in the annotation task remains low despite this. Overall, the performance of Mixtral:8x7B and Llama3.1:8B is inferior to that of smaller and faster specialized LSCD models, and Llama3.3:70B's performance for Spanish also falls short of these specialized models.

## 8 Conclusion

This study evaluated the effectiveness of large language models (LLMs) in addressing the Graded Change LSCD task, focusing on both Spanish and English datasets and comparing their performance against specifically designed and trained LSCD models that represent the state-of-the-art. In addition to evaluating model performance, we also explored the impact of automatic prompt optimization to better adapt LLMs to the task (RQ1).

A key result of this work is that LLMs, when equipped with optimized prompts, can outperform task-specific LSCD models. In particular, Llama3.3:70B, a large model paired with a prompt selected through automatic optimization, achieves state-of-the-art performance on the English dataset using the APD evaluation (RQ2). This finding highlights the potential of scaling and prompt engineering to unlock LLMs' capabilities for semantic change detection. Furthermore, Llama3.3:70B achieves the best result at the annotation level for the English dataset (RQ3). However, results across other evaluation settings are more mixed: While Llama3.1:8B and Mixtral:8x7B perform well under specific conditions (e.g., WSI-based clustering), they underperform at annotation level compared to the specialized LSCD models (RQ3). Furthermore, for Spanish, even Llama3.3:70B, despite ranking second in APD-based evaluation, fails to surpass specialized LSCD models—especially in annotation-level performance—demonstrating that challenges remain in cross-lingual generalization and fine-grained semantic interpretation.

These findings underscore the limitations of Llama3.1:8B and Mixtral:8x7B as general-purpose LLMs for the LSCD task, while also highlighting the promising potential of optimized prompts and the use of larger models for improving model performance. We will further investigate factors that may be influencing these results, including the role of quantization—for instance, whether unquantized models yield better performance on fine-grained semantic tasks. Additionally, we plan to shift our optimization target from accuracy to Spearman correlation, which better reflects the nature of graded change detection. Finally, LLMs continue to struggle with the underlying WSI problem, which may be central to improving both performance and the interpretability of results in LSCD tasks.

## Acknowledgements

This work was supported by ANID Millennium Science Initiative Program Code ICN17_002 and the National Center for Artificial Intelligence CENIA FB210017, Basal ANID.

## References

[1] J. Achiam, S. Adler, S. Agarwal, L. Ahmad, I. Akkaya, F. L. Aleman, D. Almeida, J. Altenschmidt, S. Altman, S. Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

[2] N. Arefyev, M. Fedoseev, V. Protasov, D. Homskiy, A. Davletov, and A. Panchenko. Deepmistake: Which senses are hard to distinguish for a word-in-context model. volume 2021-June, pages 16–30, 2021.

[3] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei. Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/1457c0d6bfcb4967418bfb8ac142f64a-Paper.pdf.

[4] P. Cassotti, L. Siciliani, M. DeGemmis, G. Semeraro, and P. Basile. XL-LEXEME: WiC pretrained model for cross-lingual LEXical sEMantic changE. In A. Rogers, J. Boyd-Graber, and N. Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1577–1585, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-short.135. URL https://aclanthology.org/2023.acl-short.135.

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

[6] K. D'Oosterlinck, O. Khattab, F. Remy, T. Demeester, C. Develder, and C. Potts. In-context learning for extreme multi-label classification. *arXiv preprint arXiv:2401.12178*, 2024.

[7] Q. Guo, R. Wang, G. Junliang, B. Li, K. Song, X. Tan, G. Liu, J. Bian, and Y. Yang. Connecting large language models with evolutionary algorithms yields powerful prompt optimizers. 09 2023. doi: 10.48550/arXiv.2309.08532.

[8] D. Homskiy and N. Arefyev. DeepMistake at LSCDiscovery: Can a multilingual word-in-context model replace human annotators? In *Proceedings of the 3rd International Workshop on Computational Approaches to Historical Language Change*, Dublin, Ireland, 2022. Association for Computational Linguistics.

[9] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2704–2713, 2018.

[10] A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice-Hall, Inc., 1988.

[11] A. Q. Jiang, A. Sablayrolles, A. Roux, A. Mensch, B. Savary, C. Bamford, D. S. Chaplot, D. d. l. Casas, E. B. Hanna, F. Bressand, et al. Mixtral of experts. *arXiv preprint arXiv:2401.04088*, 2024.

[12] O. Khattab, A. Singhvi, P. Maheshwari, Z. Zhang, K. Santhanam, S. Vardhamanan, S. Haq, A. Sharma, T. T. Joshi, H. Moazam, et al. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*, 2023.

[13] K. Krippendorff. *Content analysis: An introduction to its methodology*. Sage publications, 2018.

[14] A. Kutuzov and M. Giulianelli. 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, and E. Shutova, editors, *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online), Dec. 2020. International Committee for Computational Linguistics. doi: 10.18653/v1/2020.semeval-1.14. URL https://aclanthology.org/2020.semeval-1.14/.

[15] A. Kutuzov and L. Pivovarova. Rushifteval: A shared task on semantic shift detection for russian. In *Computational linguistics and intellectual technologies:*, number 20 in Komp'ûternaâ lingvistika i intellektual'nye tehnologii - Computational Linguistics and Intellectual Technologies, Russian Federation, 2021. Redkollegija sbornika. doi: 10.28995/2075-7182-2021-20-533-545. URL http://www.dialog-21.ru/en/. International Conference on Computational Linguistics and Intellectual Technologies : Dialogue 2021 ; Conference date: 16-06-2021 Through 19-06-2021.

[16] A. Kutuzov, L. Øvrelid, T. Szymanski, and E. Velldal. Diachronic word embeddings and semantic shifts: a survey. In E. M. Bender, L. Derczynski, and P. Isabelle, editors, *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA, Aug. 2018. Association for Computational Linguistics. URL https://aclanthology.org/C18-1117/.

[17] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, and G. Neubig. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. *ACM computing surveys*, 55(9):1–35, 2023.

[18] D. Loureiro, A. D'Souza, A. N. Muhajab, I. A. White, G. Wong, L. E. Anke, L. Neves, F. Barbieri, and J. Camacho-Collados. Tempowic: An evaluation benchmark for detecting meaning shift in social media. *arXiv preprint arXiv:2209.07216*, 2022.

[19] F. Martelli, N. Kalach, G. Tola, R. Navigli, et al. Semeval-2021 task 2: Multilingual and cross-lingual word-in-context disambiguation (mclwic). In *Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021)*, pages 24–36, 2021.

[20] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient Estimation of Word Representations in Vector Space. *arXiv preprint arXiv:1301.3781*, 2013.

[21] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao. Large language models: A survey. *arXiv preprint arXiv:2402.06196*, 2024.

[22] K. Opsahl-Ong, M. J. Ryan, J. Purtell, D. Broman, C. Potts, M. Zaharia, and O. Khattab. Optimizing instructions and demonstrations for multi-stage language model programs. In Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9340–9366, Miami, Florida, USA, Nov. 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.525. URL https://aclanthology.org/2024.emnlp-main.525/.

[23] T. Pasini, A. Raganato, and R. Navigli. Xl-wsd: An extra-large and cross-lingual evaluation framework for word sense disambiguation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13648–13656, 2021.

[24] T. P. Peixoto. Bayesian stochastic blockmodeling. In *Advances in Network Clustering and Blockmodeling*, chapter 11, pages 289–332. John Wiley & Sons, Ltd, 2019. doi: 10.1002/9781119483298.ch11.

[25] F. Periti and S. Montanelli. Lexical semantic change through large language models: a survey. *ACM Comput. Surv.*, 56(11), June 2024. ISSN 0360-0300. doi: 10.1145/3672393. URL https://doi.org/10.1145/3672393.

[26] F. Periti and N. Tahmasebi. A systematic comparison of contextualized word embeddings for lexical semantic change. In K. Duh, H. Gomez, and S. Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 4262–4282, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.240. URL https://aclanthology.org/2024.naacl-long.240/.

[27] F. Periti, H. Dubossarsky, and N. Tahmasebi. (chat) gpt v bert dawn of justice for semantic change detection. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 420–436, 2024.

[28] M. T. Pilehvar and J. Camacho-Collados. WiC: the word-in-context dataset for evaluating context-sensitive meaning representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1267–1273, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1128.

[29] A. Raganato, T. Pasini, J. Camacho-Collados, and M. T. Pilehvar. Xl-wic: A multilingual benchmark for evaluating semantic contextualization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7193–7206, 2020.

[30] J. Rodina and A. Kutuzov. RuSemShift: a dataset of historical lexical semantic change in Russian. In *Proceedings of the 28th International Conference on Computational Linguistics (COLING 2020)*. Association for Computational Linguistics, 2020.

[31] P. J. Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[32] D. Schlechtweg, S. Schulte im Walde, and S. Eckmann. Diachronic Usage Relatedness (DURel): A framework for the annotation of lexical semantic change. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 169–174, New Orleans, Louisiana, 2018. URL https://www.aclweb.org/anthology/N18-2027/.

[33] D. Schlechtweg, B. McGillivray, S. Hengchen, H. Dubossarsky, and N. Tahmasebi. SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain, 2020. Association for Computational Linguistics. URL https://www.aclweb.org/anthology/2020.semeval-1.1/.

[34] D. Schlechtweg, N. Tahmasebi, S. Hengchen, H. Dubossarsky, and B. McGillivray. DWUG: A large Resource of Diachronic Word Usage Graphs in Four Languages. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7079–7091, Online and Punta Cana, Dominican Republic, nov 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.emnlp-main.567.

[35] D. Schlechtweg, F. D. Zamora-Reina, F. Bravo-Marquez, and N. Arefyev. Sense through time: Diachronic word sense annotations for word sense induction and lexical semantic change detection. *Language Resources and Evaluation*, 2024.

[36] C. Spearman. The proof and measurement of association between two things. *American Journal of Psychology*, 15:88–103, 1904.

[37] N. Tahmasebi, L. Borin, and A. Jatowt. Survey of computational approaches to lexical semantic change detection. *Computational approaches to semantic change*, 6(1), 2021.

[38] X. Tang. A state-of-the-art of semantic change computation. *Natural Language Engineering*, 24, 01 2018. doi: 10.1017/S1351324918000220.

[39] U. von Luxburg. A tutorial on spectral clustering, 2007. URL https://arxiv.org/abs/0711.0189.

[40] R. Wang and M. Choi. Large language models on lexical semantic change detection: An evaluation. *arXiv preprint arXiv:2312.06002*, 2023.

[41] S. Yadav, T. Choppa, and D. Schlechtweg. Towards automating text annotation: A case study on semantic proximity annotation using gpt-4. *arXiv preprint arXiv:2407.04130*, 2024.

[42] F. D. Zamora-Reina, F. Bravo-Marquez, and D. Schlechtweg. LSCDiscovery: A shared task on semantic change discovery and detection in Spanish. In *Proceedings of the 3rd International Workshop on Computational Approaches to Historical Language Change*, Dublin, Ireland, 2022. Association for Computational Linguistics. URL https://aclanthology.org/2022.lchange-1.16/.

[43] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

[44] Y. Zhou, A. I. Muresanu, Z. Han, K. Paster, S. Pitis, H. Chan, and J. Ba. Large language models are human-level prompt engineers. In *The Eleventh International Conference on Learning Representations*, 2022.

[45] K. Zhu, J. Wang, J. Zhou, Z. Wang, H. Chen, Y. Wang, L. Yang, W. Ye, Y. Zhang, N. Zhenqiang Gong, et al. Promptbench: Towards evaluating the robustness of large language models on adversarial prompts. *arXiv e-prints*, pages arXiv–2306, 2023.