HSE at LSCDiscovery in Spanish: Clustering and Profiling for Lexical Semantic Change Discovery

Kseniia Kashleva **Alexander Shein** **Elizaveta Tukhtina** **Svetlana Vydrina***

HSE University

kkashleva@hse.ru aishein_1@edu.hse.ru eatukhtina@edu.hse.ru svvydrina@edu.hse.ru

Abstract

This paper describes the methods used for lexical semantic change discovery in Spanish. We tried the method based on BERT embeddings with clustering, the method based on grammatical profiles and the grammatical profiles method enhanced with permutation tests. BERT embeddings with clustering turned out to show the best results for both graded and binary semantic change detection outperforming the baseline.

Our best submission for graded discovery was the 3rd best result, while for binary detection it was the 2nd place (precision) and the 7th place (both F1-score and recall). Our highest precision for binary detection was 0.75 and it was achieved due to improving grammatical profiling with permutation tests.

1 Introduction

Lexical semantic change detection (LSCD) aims to identify which words and how change their meaning over time. LSCD is usually divided into two subtasks: graded change and binary change detection.

Graded LSCD is a subtask of ranking the intersection of (content-word) vocabularies according to their degree of change between a diachronic corpus pair C1 and C2 (Kurtyigit et al., 2021). In this shared task, the participants were asked to rank the set of content words in the lemma vocabulary intersection of C1 and C2 according to their degree of semantic change between C1 to C2. Submissions were scored against 60 hidden words from the full target word list which were annotated for semantic change. The total number of target words were more than 4,000 (D. Zamora-Reina et al., 2022), and, as it was a discovery task, the target words were not preselected, balanced or cleaned. Due to that, discovery is more problematic for models in comparison with semantic change detection, but it is an important task for lexicography.

Binary LSCD is a subtask of identifying whether a target word lost or gained senses from the first set of its usage to the second, or not (Schlechtweg et al., 2020).

Previous shared tasks on lexical semantic change detection (LSCD) were developed for English, German, Latin, and Swedish (Schlechtweg et al., 2020), Italian (Basile et al., 2020), and Russian (Kutuzov and Pivovarova, 2021). This one was in Spanish (D. Zamora-Reina et al., 2022). Spanish is a fusional Romance language of the Indo-European language family with rich morphology and a lot of national varieties. So far, LSCD in shared tasks were developed for three Romance languages, three German languages, and one Slavic language. Only two of them are analytical (English and Swedish), while others are fusional.

In this shared task we tested several methods. For graded change discovery we used BERT embeddings with clustering (Montariol et al., 2021). For binary change detection we used 3 methods. The first one was word embeddings again. Two others were grammatical profiling (Kutuzov et al., 2021) and grammatical profiling combined with permutation tests (Liu et al., 2021).

Though grammatical profiles by themselves yield worse performance than embedding-based method, they could be significantly improved by applying of additional significance tests.

2 Methods

2.1 BERT embeddings method

For this method[1] we used a base version of BERT with 12 attention layers and a hidden layer size of 768. The exact pre-trained model was the one for

*Equal contribution, the authors listed alphabetically.

[1]Our code is available here https://github.com/lizatukhtina/HSE-at-LSCDiscovery-in-Spanish

Proceedings of the 3rd Workshop on Computational Approaches to Historical Language Change, pages 193-198 May 26-27, 2022

Spanish[2] (Devlin et al., 2019). All parameters were set to the default as in the Transformers library, version 4.14.1 (Wolf et al., 2020).

The method consisted of several steps. First, we split the corpora into train and test sets. The train/test ratio was 90/10. We used the lemmatized version of the corpora in this method. Then we took the pre-trained BERT model for Spanish and ran a fine-tuning process on the train set of the corpora using the test set for evaluation. The code we used for fine-tuning is provided as one of the examples in the Transformers library repository.[3]

After fine-tuning the model we extracted the embeddings for the target words from the full corpora provided. The embeddings were extracted separately for two time periods. To generate a final embedding for each target word, the embeddings from all 12 attention layers of the BERT model were summarized. The embeddings for all entries of every target word were extracted this way.

As a result, we obtained two matrices for every target word. One matrix represented one time period. The dimension of the resulting matrix was Nx768, where N is the number of occurrences of the target word in the corpus of particular time period.

The final step was clustering. We ran a k-means clustering algorithm on the rows of the resulting matrices. It should be noted that we also attempted to use the affinity propagation algorithm, but it proved unfeasible at this point, as the number of target words and the number of their embeddings was too large for the affinity propagation approach. So, the final decision was to resort to the k-means algorithm which is much faster. The number of clusters was set as a hyperparameter which we tuned at the development phase. The development phase demonstrated that the results were the best when the number of clusters equaled to a multiple of 7 with the larger numbers showing better results. In order to find a balance between the clustering time and the results we decided that the number of clusters should be 28, as the larger numbers of clusters significantly increased the computational time during the prediction process. The development phase results for different numbers of clusters are shown on the Figure 1.

[2] https://huggingface.co/dccuchile/bert-base-spanish-wwm-uncased

[3] https://github.com/huggingface/transformers/tree/main/examples/pytorch/language-modeling

The resulting clusters presumably represented some gradations of word meanings. In order to calculate the graded change between the sets of clusters from two time periods, we used the average of the cosine distances between all pairs of the cluster centroids. The binary change was calculated by clustering the resulting graded changes into two clusters: the words that fall into the cluster with higher centroid value were considered as changed. The other words were considered as unchanged.

To detect binary gain/loss we took the cluster centroids for the contextualised embeddings calculated on the previous step. Those centroids were clustered once again, but this time we used the affinity propagation method that determined the number of clusters automatically. The result clusters presumably represented the basic meanings of the target words. After that we compared the number of resulting clusters for both time periods. If the number of clusters in the first period was larger than that in the second period, we assumed that this word lost a sense. If not, we assumed the word gained a sense.

As for the optional COMPARE task, our submission was identical to that for the main Graded task. We did not use any other method for that.

In terms of performance the large number of target words posed a challenge for this model during embeddings extraction and making predictions. We extracted the embeddings for all target word occurrences, so the resulting pickled file with embeddings had the size of over 40 GB. We used the HSE supercomputer cluster with 4 GPUs to parallelize our calculations (Kostenetskiy et al., 2021). The process of extracting embeddings took about 13 hours.

The process of making predictions was also slowed down by the significant number of target words. As was already mentioned above, the first attempt to use affinity propagation failed for this reason. The k-means clustering was also performed on the supercomputer. For that were used 8 CPUs and the process took approximately an hour.

This approach has the work (Montariol et al., 2021) as a foundation. We made a few changes compared to it. The first change is that we used all embeddings of the target words, while (Montariol et al., 2021) limited the number of embeddings for each word to 200. The second change is about calculating the graded change. In (Montariol et al., 2021) were used the Wasserstein distance and the

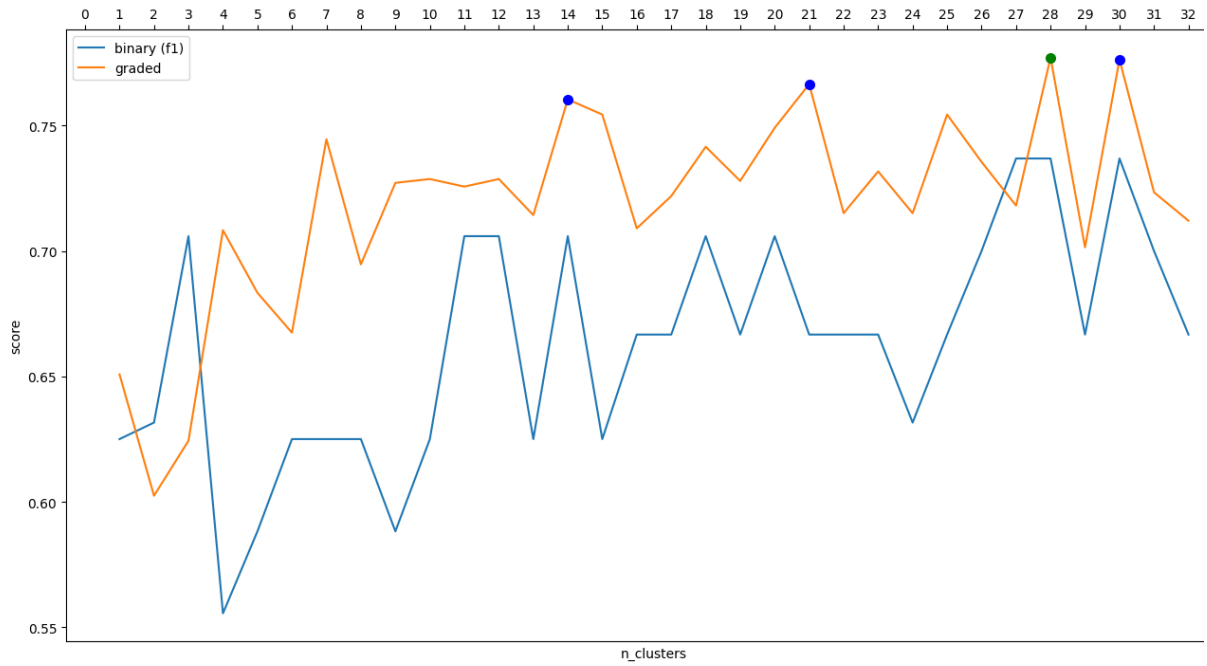

Figure 1: The figure illustrates the change in the F1-score and the Spearman rank correlation depending on the number of clusters used. The colored dots are the best results for graded change discovery. Three of them are achieved when the number of clusters is a multiple of 7. The green dot is the best number of clusters, equal to 28.

Jensen-Shannon divergence, while we used the average of all cosine distances between all cluster centroids.

2.2 Grammatical profiling

All language aspects are strongly interconnected. It means that semantic changes may be tied with grammatical changes. Diachronically, it can be observed through lexicalization and grammaticalization in particular. In Spanish, the modern usage of the verb *andar* 'to go' can be a good example of grammaticalization:

De que Blasillo ***ande*** *al escuela me e holgado mucho* (16th c.). — 'Since Blasillo has been **going** to school, I have been very happy.'

– ¿Y eso es todo el problema? — ***Ándale****, exactamente eso.* (21th c.) — 'And that's the whole problem? **Yes, yes** (lit. walk to it), that's exactly it.' (Company Company, 2008)

So here we can see that this verb changed its meaning while changing its form.

The idea of grammatical profiling is that semantic change can be discovered through significant changes in the distribution of morphosyntactic categories. This method is described in (Kutuzov et al., 2021) in detail, so here we explain only the main points. To get grammatical profiles, the frequency of morphological and syntactic categories for each target word were counted in both corpora, that were in advance tagged and parsed with UDPipe (Straka and Straková, 2017) [4]. We used raw counts for that. Then, for each target word and for both morphological and syntactic dictionaries, a list of features[5] was created by taking the union of keys in the corresponding dictionaries for the two time bins. Then, feature vectors $\vec{x_1}$ and $\vec{x_2}$ were made. Each dimension of these vectors represented a grammatical category and the value it took was the frequency of that category in the corresponding time period (Kutuzov et al., 2021). Then, the cosine distance $cos(\vec{x_1}; \vec{x_2})$ between the vectors were calculated to estimate the change in the grammatical profiles of the target word [6]. It was done separately for morphological and syntactic categories, resulting in two distance scores d_{morph} and d_{synt}. These distances can be used for graded change discovery. For binary detection, the top n target words were classified in the ranking as 'changed' (1) and others as 'stable' (0). The value of n was obtained from the ranking with the help off-the-shelf algorithms of change point detection (Truong et al., 2020).

[4] The model was *spanish-gsd-ud-2.5-191206.udpipe*

[5] These features are Universal Dependencies features https://universaldependencies.org/u/feat/index.html

[6] https://github.com/glnmario/semchange-profiling

2.3 Grammatical profiling enhanced with permutation-based statistical tests

Earlier statistical significance tests were applied to semantic change detection methods based on contextual word embeddings (Liu et al., 2021). Permutation-based statistical testing can be applied when data is limited. We used permutation tests to improve the results obtained with grammatical profiling, as the aim of the permutation test is to discover whether the observed test statistic (i.e. the cosine distance) is significantly different from zero (Liu et al., 2021). Permutation tests reassigned group labels (time periods) to all observations by sampling without replacement.

For binary change detection we calculated the default distance between grammar profiles. Then, we took sentence indices from the first and the second corpus for every target word and permute them by randomly splitting them between two time periods. If the number of possible permutations were less than 1000 we used all permutations. Then we calculated cosine distance between grammar profiles generated after shuffling. So, we have 2 sets of distances: the original cosine distance between grammar profiles and the permutated cosine distances between grammar profiles.

Let us assume, there were 5 permutations, so we got 5 distances, e.g., 0.1, 0.7, 0.4, 0.15, and 0.2, and the original cosine distance was 0.3. We took only those permutated cosine distances that were larger than the default cosine distance. In this example, these are 0.7 and 0.4 (two values). We divided the number of these larger permutated distances by the number of permutations. In this example, this is 2/5 which is a p-value (Liu et al., 2021).

If the number of permutations were greater than 1000, the procedure was the same, but we corrected the p-value for every digit capacity, i.e., we took the first significance threshold as 0.05 and step-by-step reduced it till 0.005 (Liu et al., 2021). In other words, we first randomly selected 1000 permutations and computed p-value. If this was larger 0.05, we stopped the procedure, otherwise took more permutations for more precise estimations.

As a result, we had the cosine distance between grammar profiles and the p-value for every target word. For binary change detection we sorted these values both by the distance and the p-value and labeled top n target words as changed. The coefficient n was derived with a certain set of heuristics and is subject for a further research.

3 Results

The submission results are presented in Table 1.

Clustering turned out to be the best one among all our methods. In graded change discovery it was proved to be better than both baselines and took the 3rd place in the leaderboard.

The clustering method was our only method that was applied to the optional Gain/Loss task, however, it did not show good results. While this method surpassed the baseline numbers, it proved to be significantly inferior to the other methods participating in the task. It probably happened because we approached the Gain/Loss task as a separate task. The better approach might have been to somehow use the results we received on the main Binary task in order to calculate the gain/loss values.

There is another problem with the method that we can think of. The method assigned a gain/loss label for the word if the number of clusters in two time epochs differs even by one. A better approach would probably have been to decrease the sensitivity of the method and to ignore the insignificant differences between the number of clusters.

Grammatical profiling demonstrated the worst results among three methods we used (see Table 1).

However, the results indicate that it was significantly improved by applying a permutation test. It should also be noted that grammatical profiling with a permutation test demonstrated the best precision among all participants and was only outperformed by the baseline 1. We also applied grammatical profiling for graded change discovery after the competition. The result was worse than baseline 1, but better than baseline 2 (see Table 1).

4 Discussion

Table 3 presents the top 10 words with the largest difference between BERT-based predictions and the gold standard. Closer inspection shows that there are two error types. According to the standard, some words (*actitud, banco*) changed a lot, while our prediction for these words appeared to be much lower. Meanwhile, there were words that did not change, however, our model labeled them as changed (*propiamente, fallecimiento, viernes, distribuir, variedad, socialista*). Within the top 10 words, the model fell into errors on the side of changing more often.

Table 4 presents the top 10 words with the largest difference between grammatical profiling predictions and the gold standard. Our prediction for

Binary				Gain for binary task			Loss for binary task		
Method/Team	Precision	Recall	F1	Precision	Recall	F1	Precision	Recall	F1
Baselines									
Baseline 1	0.846	0.393	0.393	—	—	—	—	—	—
Baseline 2	0.500	0.143	0.222	0.400	0.143	0.211	0.000	0.000	0.000
Our submissions: HSE team									
Clusters	0.567	**0.607**	**0.586**	0.192	0.357	0.250	0.421	0.320	0.364
Grammar	0.714	0.357	0.476	—	—	—	—	—	—
Stats	**0.750**	0.429	0.545	—	—	—	—	—	—
Best submissions of other teams									
GlossReader	0.615	0.857	0.716	0.333	0.929	0.491	0.564	0.880	0.688
DeepMistake	0.633	0.679	0.655	0.433	0.929	0.591	0.514	0.720	0.582

Table 1: Submission results for binary task: *Clusters* means embedding clustering method, *Grammar* means grammatical profiles and *Stats* means grammatical profiles combined with a permutation test. Grammatical profiling for graded discovery was made after the competition.

Graded		
Method/Team	COMPARE	Spearman
Baselines		
Baseline 1	0.561	0.543
Baseline 2	0.088	0.092
Our submissions: HSE team		
Clusters	0.558	0.553
Grammar	—	*0.390*
Best submissions of other teams		
GlossReader	0.842	0.735
DeepMistake	0.829	0.702

Table 2: Submission results for graded task. Grammatical profiling was made after the competition.

these words was much lower than the gold standard. Some incorrect predictions are the same with the incorrect predictions obtained with the BERT-based method (*actitud, canal, banco*). A likely explanation is that these words have a complicated semantic structure and more than one meaning.

5 Conclusion

Further studies need to be carried out in order to evaluate the combination of profiling with statistical significance testing for other languages. The great advantage of grammatical profiling is that computational resources required for that method are quite low. It is helpful when the number of target words is great, like in this shared task for graded discovery.

Although the BERT-based method demonstrated the best results, more detailed error analysis is still required.

word	change graded	change graded golden	change graded difference
actitud	0.369	0.925	0.556
propiamente	0.473	0	0.473
fallecimiento	0.468	0	0.468
viernes	0.447	0	0.447
trato	0.490	0.051	0.439
distribuir	0.438	0	0.438
banco	0.514	0.925	0.411
canal	0.607	1	0.393
variedad	0.392	0	0.392
socialista	0.391	0	0.391

Table 3: BERT-based predictions compared with the gold standard.

word	change graded	change graded golden	change graded difference
marco	0.018	1	0.982
prima	0.118	1	0.882
actitud	0.115	0.925	0.810
indicativo	0.202	1	0.798
canal	0.240	1	0.760
disco	0.167	0.915	0.748
pendiente	0.096	0.781	0.685
corriente	0.072	0.753	0.681
banco	0.246	0.925	0.678
cólera	0.098	0.741	0.643

Table 4: Grammatical profiles predictions compared with the gold standard.

Acknowledgements

This research was supported in part through computational resources of HPC facilities at HSE University (Kostenetskiy et al., 2021).

References

Pierpaolo Basile, Annalina Caputo, Tommaso Caselli, Pierluigi Cassotti, and Rossella Varvara. 2020. Diacr-ita @ evalita2020: Overview of the evalita2020 diachronic lexical semantics (diacr-ita) task. In *EVALITA*.

Concepción Company Company. 2008. The directionality of grammaticalization in spanish. *Journal of Historical Pragmatics*, 9(2):200–224.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. 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. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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. Association for Computational Linguistics.

P. S. Kostenetskiy, R. A. Chulkevich, and V. I. Kozyrev. 2021. HPC resources of the higher school of economics. *Journal of Physics: Conference Series*, 1740(1):012050.

Sinan Kurtyigit, Maike Park, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2021. Lexical semantic change discovery. In *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)*, pages 6985–6998, Online. Association for Computational Linguistics.

Andrey Kutuzov and Lidia Pivovarova. 2021. Three-part diachronic semantic change dataset for Russian. In *Proceedings of the 2nd International Workshop on Computational Approaches to Historical Language Change 2021*, pages 7–13, Online. Association for Computational Linguistics.

Andrey Kutuzov, Lidia Pivovarova, and Mario Giulianelli. 2021. Grammatical profiling for semantic change detection. In *Proceedings of the 25th Conference on Computational Natural Language Learning*, pages 423–434, Online. Association for Computational Linguistics.

Yang Liu, Alan Medlar, and Dorota Glowacka. 2021. Statistically significant detection of semantic shifts using contextual word embeddings. In *Proceedings of the 2nd Workshop on Evaluation and Comparison of NLP Systems*, pages 104–113, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Syrielle Montariol, Matej Martinc, and Lidia Pivovarova. 2021. Scalable and interpretable semantic change detection. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4642–4652, Online. Association for Computational Linguistics.

Dominik Schlechtweg, Barbara McGillivray, Simon Hengchen, Haim Dubossarsky, and Nina Tahmasebi. 2020. SemEval-2020 task 1: Unsupervised lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 1–23, Barcelona (online). International Committee for Computational Linguistics.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99, Vancouver, Canada. Association for Computational Linguistics.

Charles Truong, Laurent Oudre, and Nicolas Vayatis. 2020. Selective review of offline change point detection methods. *Signal Processing*, 167:107299.

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