BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection

Nikolay Arefyev
Lomonosov Moscow State University,
Samsung R&D Institute Russia,
HSE University
Moscow, Russia
nick.arefyev@gmail.com

Vasily Zhikov
Lomonosov Moscow State University
Moscow, Russia
zhikovn11@gmail.com

Abstract

SemEval-2020 Task 1 is devoted to detection of changes in word meaning over time. The first subtask raises a question if a particular word has acquired or lost any of its senses during the given time period. The second subtask requires estimating the change in frequencies of the word senses. We have submitted two solutions for both subtasks. The first solution performs word sense induction (WSI) first, then makes the decision based on the induced word senses. We extend the existing WSI method based on clustering of lexical substitutes generated with neural language models and adapt it to the task. The second solution exploits a well-known approach to semantic change detection, that includes building word2vec SGNS vectors, aligning them with Orthogonal Procrustes and calculating cosine distance between resulting vectors. While WSI-based solution performs better in Subtask 1, which requires binary decisions, the second solution outperforms it in Subtask 2 and obtains the 3rd best result in this subtask.

1 Introduction

Lexical semantic change detection (LSCD) is a problem of detecting changes in word meaning over time. SemEval-2020 Task 1 (Schlechtweg et al., 2020) suggests two variations of this problem. The first one (Subtask 1) requires detection of words that have changed the set of their senses, either acquiring a new sense or loosing an old one between two given time periods. For this subtask we extend and adapt the state-of-the-art WSI method based on lexical substitution with neural language models proposed in (Amrami and Goldberg, 2019). Instead of the English BERT employed by the original method, we use the recently introduced masked language model XLM-R (Conneau et al., 2019), which was trained on all languages considered in the task. We adapt this model to the task datasets by additionally finetuning it to accept lemmatized texts as input and produce lemmatized substitutes. We also experiment with different dynamic patterns and their combinations, as well as multitoken substitutes generation. For Subtask 2 we obtained better results with another approach based on SGNS word embeddings and Orthogonal Procrustes alignment.

2 Related work

In this section we describe those works, which our solutions are directly based on. Please, refer to (Schlechtweg et al., 2020) for a more detailed overview of the task and alternative approaches.

Subtask 1 raises the question if the set of senses of a particular word has changed. This task would be simple to solve, if we could identify word senses and corresponding word occurrences first. WSI is a task of clustering occurrences of an ambiguous word in accordance to its senses. One of our solutions employs a substitution-based approach (Baskaya et al., 2013; Amrami and Goldberg, 2018; Arefyev et al., 2019), which exploits lexical substitutes to distinguish word senses. In the recently proposed state-of-the-art implementation of this approach (Amrami and Goldberg, 2019), for each occurrence of an ambiguous word lexical substitutes (i.e. words that can replace the ambiguous word in a given context) are generated

Proceedings of the 14th International Workshop on Semantic Evaluation, pages 171–179
Barcelona, Spain (Online), December 12, 2020.

using BERT masked language model (MLM) (Devlin et al., 2018). Then substitute vectors are built, which are TF-IDF weighted bag-of-words vectors based not on the words in context, but on the generated substitutes. Finally, these vectors are clustered using agglomerative clustering and the resulting clusters are taken as induced senses.

XLM-R (Conneau et al., 2019) is an MLM trained similarly to the multilingual version of BERT on texts in 100 languages (including four languages considered in the shared task), but having 3x more parameters, 2x larger vocabulary and trained not only on Wikipedia, but also on Common Crawl dataset, which increased training data for low-resourced languages by orders of magnitude resulting in more than 2TB of data in total. XLM-R has shown better performance than multilingual BERT in different NLP tasks. Based on these advantages, we have decided to employ XLM-R for substitutes generation in our solution.

Subtask 2 raises the question of how much the frequency distribution of senses has changed for a particular word. This task is well explored and can be solved without explicit WSI by exploiting implicit features like the change in frequencies of word contexts. The surveys like (Kutuzov et al., 2018), (Tahmasebi et al., 2018) and (Schlechtweg et al., 2019) present the comparison of different approaches to this task. Most of these approaches, including (Dubossarsky et al., 2019) (Hamilton et al., 2016), imply using Skip-gram Negative Sampling (SGNS) word representations and computing cosine distances between them. We employed one of such methods as our baseline solution.

3 Our Solutions

3.1 SGNS+OP+CD

As a baseline solution we used SGNS vectors with Orthogonal Procrustes alignment and cosine distance (SGNS+OP+CD), since this combination is known to have strong performance for a task similar to Subtask 2 (Schlechtweg et al., 2019)[1]. We did not tune hyperparameters, instead we used one of the configurations listed in the paper: *window_size*=5, *k*=5, *t*=0.001, *dimensions*=300, *minCount*=0, *iters*=5. To solve task 1, we predicted positive class if the cosine distance was above a certain threshold, which was set to 0.5, because no information about class proportions was available. After the competition we have analysed the effect of the threshold value on the results, see figure 1.

3.2 BOS+AggloSil+DC

Our WSI-based solution employs a WSI method that is being developed in our parallel work on WSI. It involves the following steps. For a particular target word its occurrences in old and new corpora are collected, and lexical substitutes are generated for each of them. Then WSI is performed by clustering bag-of-substitutes (BOS) vectors with agglomerative clustering employing silhouette score to select the number of clusters (AggloSil). Finally, we search for a decision cluster (DC), i.e. a cluster that has large number of occurrences from one corpus and small from another, and predict semantic change if such cluster exists. Next we provide detailed description of each step.

Lexical substitutes generation. We exploit the multilingual masked language model XLM-R (Conneau et al., 2019) to generate lexical substitutes for a given occurrence of some target word. The simplest option is replacing the target word in the given text fragment with a special token "<mask>" and passing this modified text to XLM-R, which is trained to generate words that can appear in masked positions. However, this usually results in substitutes that are not related to the target. Following (Amrami and Goldberg, 2019), we experimented with dynamic patterns. For instance, we may replace the target with "**T and < mask>**" and then replace T back with the target (this pattern is denoted "**T and M**" for brevity). Thus, given a sentence *I love old planes* with the ambiguous word *planes*, instead of *I love old <mask>*, the model receives *I love old planes and <mask>*. In the latter case, the model returns substitutes like *vehicles, machines, stuff,* etc., which are more closely related to the target, than substitutes returned in the former case like *times, school, movies,* etc. Since XLM-R was trained on raw texts while the task corpora are pre-processed and lemmatized, we finetuned it using the MLM objective on a small subset of

[1]We used the implementation that accompanies the aforementioned paper: https://github.com/Garrafao/LSCDetection

72K randomly sampled sentences with equal number of sentences from old and new corpora for each language. This accustoms XLM-R to work with pre-processed and lemmatized inputs, as well as return lemmas instead of word forms as substitutes. Finetuning on the whole task corpora may improve results, but requires much more computational resources, hence, we leave it for the future work.

In the post-evaluation phase we experimented with multi-subword substitutes such as "**T and MM**". This means inserting several masks at input, predicting $topk$ most probable fillers for the first one and one most probable continuation for each of them (we also tried beam search, which gave no improvements). Additionally, we tried combining symmetric patterns, meaning that the probabilities of substitutes for the patterns "**T and MM**" and "**MM and T**" are multiplied before selecting the most probable substitutes.

Bag-of-substitutes (BOS) vectors. For each occurrence of a target word we only take $topk$ substitutes with the highest probabilities. We suppose that the target word has multiple senses and thus it cannot be useful as a substitute, so we remove it from the substitutes. After that we filter substitutes that were generated for less than min_df or more than max_df fraction of occurrences of the same target, since too rare or too frequent substitutes are likely to be useless for discriminating between senses of this target. Finally, bag-of-substitutes vectors are built which are basically bag-of-words vectors for substitutes, not for the original text fragments.

Additionally, substitutes can be lemmatized. This was found useful for the default (non-finetuned) XLM-R language model, because it tends to generate substitutes in different grammatical forms depending on the context, thus increasing sparsity (Amrami and Goldberg, 2018). However, our MLM is finetuned to predict lemmas, so it is less important in our case.

Clustering. For each target word separately we cluster all BOS vectors of its occurrences from the old and the new corpora together. Following (Amrami and Goldberg, 2019) we use the agglomerative clustering algorithm with cosine distance and average linkage. The number of clusters that maximizes the silhouette score[2] of clustering is selected.

Final prediction. The decision function returning the final binary label for Subtask 1 is based on the labeling criterion provided by the task organizers (Schlechtweg et al., 2020). Namely, if there is a cluster containing less than k examples from the old or the new corpus and more than n examples from another one (we call it decision cluster, or DC), then we predict that the change of word senses took place.

To solve Subtask 2, for each target we build two vectors having dimensionality equal to the number of clusters and containing the number of examples from the old or the new corpora in each cluster. The final score is the cosine distance between these vectors.

4 Experiments and Results

4.1 Datasets

All our results reported below were obtained on the test sets presented in the competition SemEval-2020 Task 1 (Schlechtweg et al., 2020). They are based on English (Alatrash et al., 2020), German (Deutsches Textarchiv, 2017; Berliner Zeitung, 2018), Swedish (Språkbanken, Downloaded in 2019) and Latin (McGillivray and Kilgarriff, 2013) corpora. Since no train or development sets were provided, we employed WSI and LSCD datasets for the Russian language, namely, bts-rnc (Panchenko et al., 2018) and macro (Fomin et al., 2019), to select good hyperparameters while not overfitting to the test sets.

4.2 Subtask 1: Binary Classification

Our results during evaluation and post-evaluation periods for Subtask 1 are reported in tables 1 and 2. Since no estimates of class proportions were available during the evaluation period, we used the threshold of 0.5 in our SGNS+OP+CD solution for Subtask 1, meaning that we predict that the semantic change happened if the cosine distance is larger than 0.5. Figure 1 shows how the accuracy of the SGNS+OP+CD approach dramatically depends on the choice of this threshold. At the border points of the plot (0.0 and

[2] https://scikit-learn.org/stable/modules/generated/sklearn.metrics.silhouette_score.html

1.0) the accuracy equals to the proportions of examples that belong to the positive or the negative class respectively. As we can see, SGNS+OP+CD accuracy with the selected threshold is only a little bit higher than the most frequent class (MFC) classifier accuracy on English and German, and is significantly lower on Latin and Swedish. The optimal threshold depends on the language and its performance is not much better than that of the MFC. Moreover, according to the table 1 the MFC accuracy is comparable to the best participants' results on Latin and Swedish. Hence, for our methods we report macro-averaged F1 score along with accuracy, which is the official metric. Additionally, it worth noting that due to the small number of test examples (30-50 words per language), one should be cautious when drawing any conclusions from the observed results. In particular, we tried to check the statistical significance of the difference between our results and the best result for each language using McNemar's test and Wilcoxon signed-rank test [3], but both tests failed to reject the null hypothesis at the significance level of 0.05 for all languages except German.

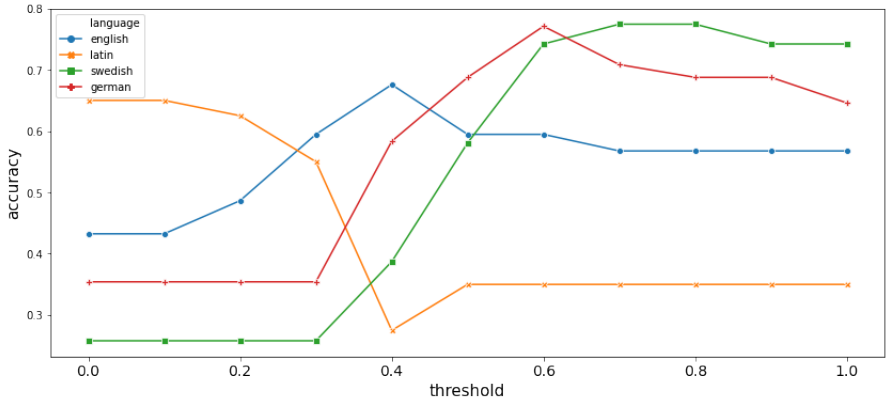

Figure 1: Accuracy of SGNS+OP+CD in Subtask 1 depending on the threshold value.

During the evaluation period we did not select the optimal hyperparameters of the WSI method due to the lack of time, instead all submissions using the BOS approach share most of the hyperparameters which were set intuitively. We used the dynamic pattern "**M or T**", $topk$=500, TF-IDF vectorizer with max_df=0.98. The number of clusters was picked between 2 and 13 by silhouette score. The thresholds n, k were set to the values 0,1 for Latin and 2,5 for other languages, which are specified in the task description. The only hyperparameter we varied was min_df, which is the minimal proportion of examples a particular substitute should be generated for to survive after filtering. According to our previous experience, it is crucial to select this hyperparameter, because it allows to filter out noisy substitutes while preserving useful ones. Table 1 compares our submissions during the evaluation phase to the winners of the competition. For BOS+AggloSil+DC method the best (v.3) among three submissions is shown, with min_df set to 0.02 for Swedish and 0.01 for other languages.

In the post-evaluation experiments we switched to the hyperparameters selected on the Russian WSI dataset: the dynamic pattern "**MM or T**" combined with the symmetric one "**T or MM**", $topk$=150, count vectorizer with min_df=0.03, max_df=0.8. We have noticed that small n, k result in many false positives. Hence, we decided to use values that are optimal for the Russian LSCD dataset, which are 10,15. Selected hyperparameters improved the results for all languages except Latin (where the positive class dominates, hence, more conservative thresholds are harmful).

From table 2 we can see that without dynamic patterns substitutes consisting of two subwords ("**MM**") often outperform single subword substitutes ("**M**"). We cannot reliably tell whether the usage of the dynamic patterns is effective or not. The target words are mostly nouns and Amrami and Goldberg (2018) shows that symmetric patterns have relatively small impact on WSI quality for nouns. The finetuned model seems to perform better for all languages except Swedish (for Swedish accuracy is the same, but F1 score

[3] as implemented in `https://www.statsmodels.org/dev/generated/statsmodels.stats.contingency_tables.mcnemar.html` and `https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.wilcoxon.html`, with Pratt modification for zero differences

System	English		German		Latin		Swedish	
	acc	F1	acc	F1	acc	F1	acc	F1
MFC baseline	0.568	n/a	0.646	n/a	0.650	n/a	0.742	n/a
top evaluation phase submissions								
UWB	0.622	-	**0.750**	-	**0.700**	-	0.677	-
Life-Language	**0.703**	-	**0.750**	-	0.550	-	**0.742**	-
Jiaxin & Jinan	0.649	-	0.729	-	**0.700**	-	0.581	-
our submissions*: BOS+AggloSil+DC (team cs2020), and SGNS+OP+CD (cs2021)								
cs2020, BOS+AggloSil+DC (v.3)	0.595	0.575	0.500	0.438	0.575	0.525	0.677	0.609
cs2021, SGNS+OP+CD	0.595	0.427	0.688	0.286	0.400	0.564	0.581	0.684

Table 1: Subtask 1 evaluation phase results. *Due to problems with the submission platform, we could not add users to a single team. Hence, we created a new one, but have made only four submissions in total.

System	English		German		Latin		Swedish	
	acc	F1	acc	F1	acc	F1	acc	F1
"MM or T", comb, finetuned	0.649	0.590	0.646	0.514	0.350	0.259	**0.742**	0.426
min_df 0.03 → 0.01	0.595	0.527	0.688	0.571	0.375	0.301	**0.742**	**0.523**
min_df 0.03 → 0.02	0.622	0.568	0.646	0.482	0.325	0.245	**0.742**	0.426
n,*k* from task spec*	0.622	0.52	0.583	0.368	**0.55**	**0.4**	0.710	0.415
- finetuned	0.595	0.527	0.625	0.469	0.350	0.259	**0.742**	**0.523**
- comb	0.568	0.413	0.688	0.571	0.350	0.259	**0.742**	0.426
or → *and*	0.541	0.435	0.646	0.514	0.350	0.259	0.710	0.415
+ lemm	0.595	0.469	**0.708**	**0.611**	-	-	-	-
- finetuned + lemm	0.568	0.413	0.667	0.528	-	-	-	-
"M or T", comb, finetuned	**0.703**	0.634	0.625	0.564	0.375	0.324	0.701	0.503
- comb	**0.703**	**0.653**	0.625	0.525	0.375	0.355	**0.742**	**0.523**
"MM", finetuned	0.595	0.427	0.667	0.556	0.325	0.245	**0.742**	0.426
"M", finetuned	0.568	0.483	0.604	0.486	0.325	0.289	0.677	0.404

Table 2: Subtask 1 post-evaluation results. *We change (*n*,*k*) from (10,15) to (0,1) for Latin and (2,5) for other languages.

is lower). The "**MM and T**" pattern seems to perform worse than the default one "**MM or T**" (except for German where the patterns perform similarly). If dynamic patterns are used, combining symmetric patterns or generating two subword substitutes did not show consistent improvements. Moreover, the "**M or T**" pattern with no patterns combination outperformed the default method on most languages (except German). This contradicts our results in WSI for the Russian language, where these techniques significantly improve WSI performance. One possible explanation is lemmatized datasets, which may negatively effect XLM-R performance even after finetuning. To check this, we have lemmatized Russian WSI dataset and found large drop in performance of WSI when non-finetuned XLM-R is used. Finetuned XLM-R performed much better, however still worse than non-finetuned version on non-lemmatized raw texts. The optimal values of *k* and *n* parameters appear to vary vastly for different languages and datasets. For instance, after changing them from the values selected on the Russian LSCD dataset to the values specified by the organizers, the results improved for Latin, but worsened for all other languages.

4.3 Subtask 2: Ranking

System	English	German	Latin	Swedish
top evaluation phase submissions				
mpoemsl (1st)	**0.422** (4)	**0.725** (2)	0.412 (20)	0.547 (5)
jinan (2nd)	0.325 (21)	0.717 (3)	**0.440** (13)	**0.588** (2)
our submissions (team cs2021)				
SGNS+OP+CD (3rd)	0.375 (9)	0.702 (5)	0.399 (22)	0.536 (7)
post-evaluation results				
BOS+AggloSil+CD	0.299	0.094	-0.134	0.274

Table 3: Subtask 2 results, Spearman rank correlation.

On the second subtask the SGNS+OP+CD approach showed pretty good performance, especially on German and Swedish languages. Overall, we obtained 3rd best result. The results of the BOS+AggloSil+CD approach however are not that great. We assume that it is due to inappropriate number of clusters often selected by silhouette (in several cases there were only two clusters). The results are shown in table 3.

4.4 Decision Process Analysis

One of the advantages of the BOS+AggloSil+DC method is a possibility to explain its decision for a particular target word by analyzing clusters of occurrences of this word and corresponding substitutes. We have developed a web application, that visualizes the process of solving WSI or LSCD task by our system[4]. This application can be useful for error analysis and potentially can be helpful to linguists for their research in lexical semantics. In table 4 we provide examples of information from the application explaining system decisions. For each cluster we display the most probable substitutes for old / new examples. The probability is estimated as the proportion of examples from old / new corpus, which a particular substitute was generated for. Table 4 shows 7 most probable substitutes for the largest cluster (LC), often corresponding to the most frequent sense, and the decision cluster (DC).

Word	Cluster	Substitutes for examples from old / new corpus	Example counts
Player	LC	thief, soldier, gamer, singer, clerk, comedian, composer / striker, receiver, pitcher, commentator, scorer, gamer, squad	111 / 761
	DC	bee, photographer, competitor, mute, commentator, shooter, ladder / library, recorder, receiver, tracker, mixer, cassette, desk	1 / 46
Ounce	LC	tonne, punch, flour, dime, bean, collar, shelf / dime, punch, collar, tonne, bean, lb, slice	98 / 142
	DC	twelve, quartz, 8, fewer, iron, forty, hefty / ounce, 4, 8, 5, 3, 2, 12	3 / 24

Table 4: System decision explanation examples. Substitutes were obtained using the combination of "**T or MM**" and "**MM or T**" patterns, target words were not removed.

The first example (*player*) appears to gain the new sense "media player", since there is a cluster with almost all examples from the new corpus, in which frequent substitutes for the target word are *receiver*, *recorder*, etc. The second example (*ounce*) is an example of incorrect prediction of our system. The decision cluster includes only examples with the word *ounce* preceded by a number, such as *three-ounce*, *10-ounce*, etc. Thus the language model mostly generates numbers as substitutes. This example shows that the specific cases of word usages can destabilize the performance of our system. For both words the decision cluster included only insignificant amount of examples from old corpus. Thus the top substitutes for old corpus examples in the decision cluster are irrelevant.

5 Conclusion

We proposed the WSI-based solution for the lexical semantic change detection problem. It outperforms the well-known SGNS+OP+CD baseline on Subtask 1. However, the baseline performs better for Subtask 2, where we obtained the 3rd best result. We have shown that finetuning of XLM-R on the lemmatized task corpora improves the final results for Subtask 1. A comparison of different variants of our method is provided. However, a larger dataset is required to draw reliable conclusions from these observations.

Acknowledgements

We thank the organizers of the competition for such an inspiring task. We are grateful to our reviewers for their useful suggestions. The contribution of Nikolay Arefyev to the paper was partially done within the framework of the HSE University Basic Research Program funded by the Russian Academic Excellence Project '5-100'.

[4]The code for reproducing the results of BOS+AggloSil+DC method and visualizing them is available here: `https://github.com/DeadBread/BOS_AggloSil`

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. Ccoha: Clean corpus of historical american english. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 6958–6966.

Asaf Amrami and Yoav Goldberg. 2018. Word sense induction with neural biLM and symmetric patterns. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4860–4867, Brussels, Belgium, October-November. Association for Computational Linguistics.

Asaf Amrami and Yoav Goldberg. 2019. Towards better substitution-based word sense induction. *CoRR*, abs/1905.12598.

Nikolay Arefyev, Boris Sheludko, and Alexander Panchenko. 2019. Combining lexical substitutes in neural word sense induction. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing (RANLP'19)*, RANLP '19, pages 62–70, Varna, Bulgaria.

Osman Baskaya, Enis Sert, Volkan Cirik, and Deniz Yuret. 2013. AI-KU: Using substitute vectors and co-occurrence modeling for word sense induction and disambiguation. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 300–306, Atlanta, Georgia, USA, June. Association for Computational Linguistics.

Berliner Zeitung. 2018. Diachronic newspaper corpus published by Staatsbibliothek zu Berlin.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Unsupervised cross-lingual representation learning at scale. *CoRR*, abs/1911.02116.

Deutsches Textarchiv. 2017. Grundlage für ein Referenzkorpus der neuhochdeutschen Sprache. Herausgegeben von der Berlin-Brandenburgischen Akademie der Wissenschaften.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. cite arxiv:1810.04805Comment: 13 pages.

Haim Dubossarsky, Simon Hengchen, Nina Tahmasebi, and Dominik Schlechtweg. 2019. Time-out: Temporal referencing for robust modeling of lexical semantic change. *arXiv preprint arXiv:1906.01688*.

Vadim Fomin, Daria Bakshandaeva, Julia Rodina, and Andrey Kutuzov. 2019. Tracing cultural diachronic semantic shifts in russian using word embeddings: test sets and baselines. *CoRR*, abs/1905.06837.

William L Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. *arXiv preprint arXiv:1605.09096*.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: a survey. *arXiv preprint arXiv:1806.03537*.

Barbara McGillivray and Adam Kilgarriff. 2013. Tools for historical corpus research, and a corpus of latin. *New Methods in Historical Corpus Linguistics*, (3):247–257.

Alexander Panchenko, Anastasiya Lopukhina, Dmitry Ustalov, Konstantin Lopukhin, Nikolay Arefyev, Alexey Leontyev, and Natalia Loukachevitch. 2018. Russe'2018: a shared task on word sense induction for the russian language. *Computational Linguistics and Intellectual Technologies*, pages 547–564.

Dominik Schlechtweg, Anna Hätty, Marco Del Tredici, and Sabine Schulte im Walde. 2019. A wind of change: Detecting and evaluating lexical semantic change across times and domains. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 732–746, Florence, Italy, July. 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 *To appear in Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Språkbanken. Downloaded in 2019. The Kubhist Corpus. Department of Swedish, University of Gothenburg.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2018. Survey of computational approaches to lexical semantic change. *arXiv preprint arXiv:1811.06278*.

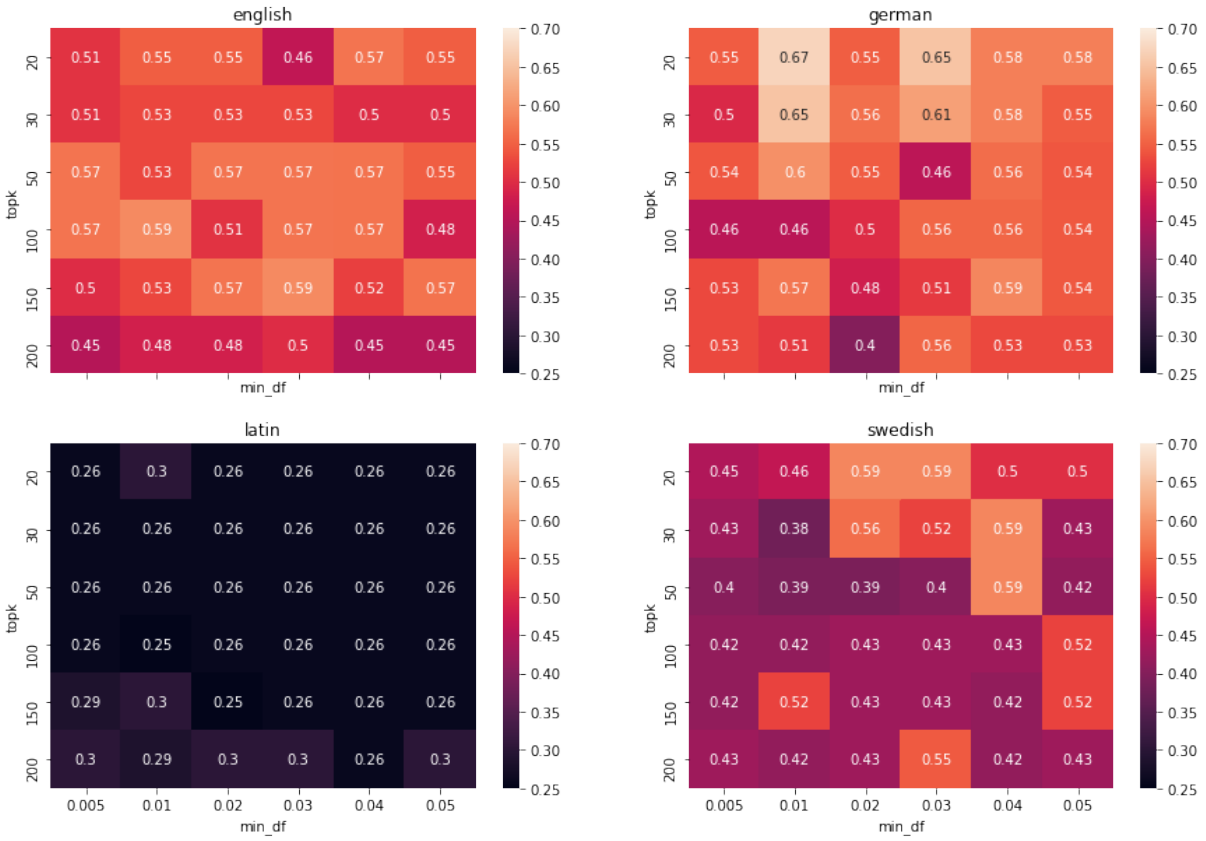

Figure 2: Subtask 1 Macro-F1 score for different values of $topk$ and min_df.

A Subtask 1 results depending from hyperparameters

In this section we show the dependence of macro-F1 score for subtask 1 from values of hyperparemeters. Figure 2 shows the effect of $topk$ and min_df on the performance. The rest hyperparameters are the same as in first line of table 1. Notice, that the variance of the results is very high: even small change in hyperparameter values may result in F1 score changing by more than 10 points, which is due to the small test set size. Due to this variance, even selecting hyperparameters on a development set of the same language, if such dataset exists, will likely be useless.

Figure 3 shows the dependence of performance on the thresholds of the final decision step. We see clear dependence from k for English and German. This is due to imperfect WSI predictions that often put in each cluster a few examples of occurrences with unrelated word senses. Larger values of k allow the model to ignore this noise.

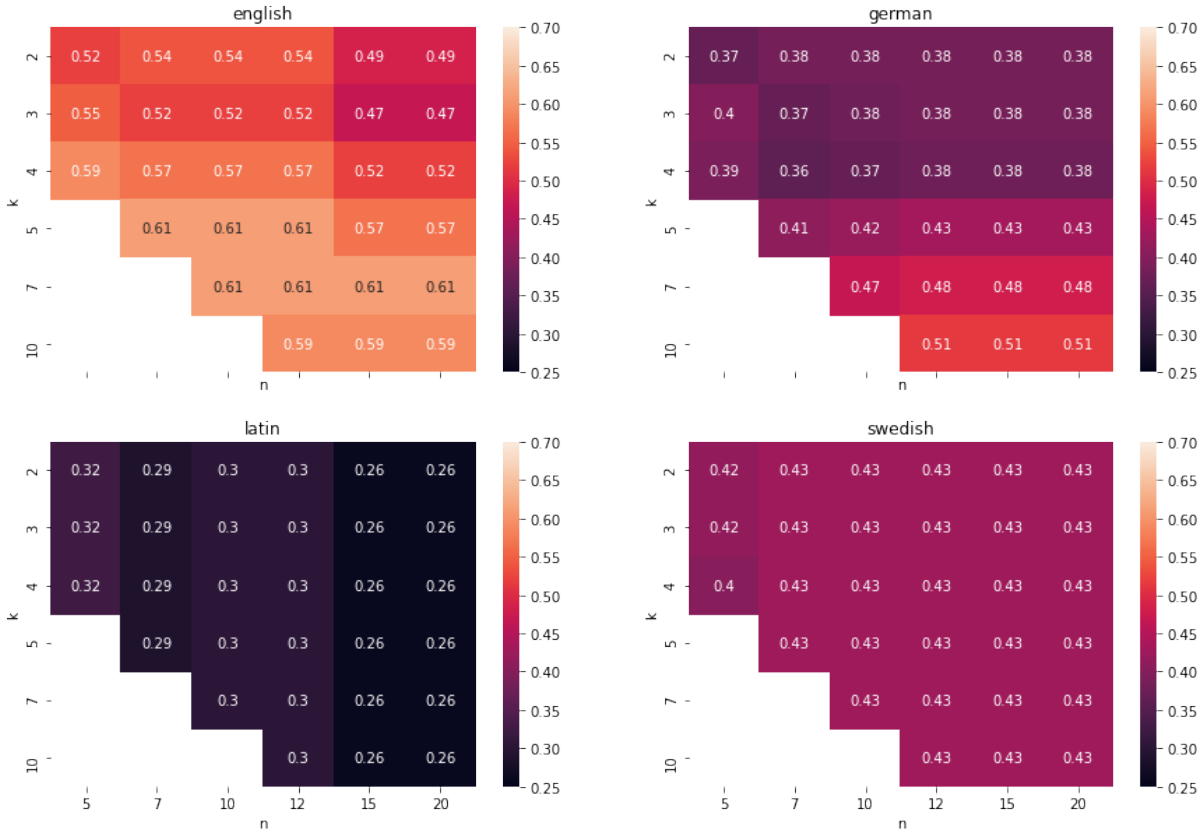

Figure 3: Subtask 1 Macro-F1 score for different values of k,n thresholds.