The Finer They Get: Combining Fine-Tuned Models For Better Semantic Change Detection

Anonymous Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Anonymouser Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Anonymousest Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

In this work we propose enriching contextualized models using a range NLP and linguistic fine-tuning tasks to improve the capacity of capturing lexical semantic change. We include tasks aimed to capture both low-level linguistic information like part-of-speech tagging, as well as higher-level (semantic) information.

Through a series of analyses we show that certain combinations of fine-tuning tasks, like sentiment, syntactic information, and logical inference, bring large improvements to standard LSC models that are based only on standard language modeling. We test on the binary classification and ranking tasks of SemEval-2020 Task 1 and evaluate using permutation tests.

1 Introduction

The last few years have seen a growing interest in language change, specifically in lexical semantic change (LSC), from the NLP community. LSC is the linguistic phenomenon in which words' meaning may change over time (for example by adding senses, broadening, or narrowing in scope). Originally, the study of lexical semantic change aspired to understand the phenomenon from a linguistic perspective (Dubossarsky et al., 2016; Schlechtweg et al., 2017; Keidar et al., 2022). It was also motivated by the need for better handling of semantic change in other text-based research disciplines that work with historical texts (e.g., lexicographers, historians).

In more recent times, the understanding that general purpose NLP models also need to accommodate for the fluidity of word meaning has reached the greater NLP community, bringing with it the realization that LSC plays a vital role (Barbieri et al., 2022). It is particularly visible in the deterioration of model performance over time because the language on which models and other algorithms are (pre-)trained, starts to drift as time passes (Rosin and Radinsky, 2022; Alkhalifa et al., 2023). When deployed, these models process text from time periods they were not trained on, which hinder their performance.

This wide, multi-disciplinary interest in LSC, has driven the development of many computational models for language change (Tahmasebi et al., 2021; Kutuzov et al., 2018). In addition, much work has been devoted to supporting this progress by curating evaluation datasets that provide appropriate testing of these new models. Most of these datasets are in the form of SemEval tasks and contain high quality, humanly annotated lists of words. Each word has either changed its meaning between the considered time periods, or remained stable in meaning. Each list is accompanied by relevant historical corpora and has become the de-facto evaluation standard in the field.

When reviewing the different models that are evaluated in SemEval, most of them use the same suit of methods that rely on standard distributional models of meaning. These models are either trained solely on historical text (e.g., SGNS or other static models), or use contextualized models pre-trained on a large 'general purpose' text[1]. All of these models generate meaning representations in vector form for words from a historical corpus and compare them to vectors representing the modern meaning of the same word. Although the models differ (e.g., in terms of data or training objectives), they all share the same basic trait - they rely on meaning representation learned only from a Masked Language Modeling task without additional linguistic information.

This state of affairs raises a question: do models that are based solely on collocation statistics and

[1] Some contextualized models are also fine-tuned on historical corpora.

trained exclusively on a masked-word prediction task suffice for our purpose? That is, are classic distributional models able to capture the full repertoire of word meaning, and then access it when analyzing meaning change? In this work, we investigate several concerns that we think a-priori suggest that additional linguistic information is beneficial for unsupervised lexical semantic change detection.

One direct path forward is fine-tuning pre-trained models on additional tasks or domains like Question Answering and Sentiment Analysis. It has been demonstrated that fine-tuning of models helps even when the fine-tuned tasks are different than the target task (for example, fine-tuning on textual summarization and testing on Question-Answering) in a sort of a zero-/few-shot transfer learning (Peters et al., 2019; Merchant et al., 2020; Khashabi et al., 2020). Therefore, it is reasonable to assume that enriching a contextualized model with additional fine-tuning tasks would lead to improved performance also for LSC detection.

In this paper, we test this hypothesis with one of the top-performing models of LSC detection in English and explore the potential to improve it by enriching it through a set of fine-tuning tasks.

2 Related literature

2.1 Models of LSC and their evaluation

In the past years, we have seen an increasing amount of models for unsupervised detection of lexical semantic change, almost exclusively focused on distributional semantic models. SemEval-2020 Task 1 was the first attempt at a large-scale evaluation and comparison of methods on four different languages. Two main classes of methods were evaluated by the participating teams. The first was based on type embeddings, either those that require alignment between independently trained models, (e.g., SGNS with Orthogonal Procrustes alignment (Arefyev and Zhikov, 2020)) or static embeddings without the requirement of alignment (Zamora-Reina and Bravo-Marquez, 2020). The second class was based on contextualized embeddings and combined with e.g., a clustering algorithm to derive sense information (e.g., XLM-R with K-means clustering (Gyllensten et al., 2020)) or other means of comparing 100 vectors in each time period with each other (Kutuzov and Giulianelli, 2020). The models were evaluated on two tasks, binary classification and ranking.

For the four SemEval datasets, the trend was that type-based models outperformed contextualized (token) models. It was also clear that good performance on binary classification does not necessarily indicate good performance on the ranking task. Since SemEval-2020, more evaluation tasks have been curated for Russian (Kutuzov et al., 2021), Italian (Basile et al., 2020), Norwegian (Kutuzov et al., 2022) and Spanish (Zamora-Reina et al., 2022), where we see stronger indications that contextualized models perform better than type-based ones.

2.2 Contextualized models training

Training contextualized models requires a massive amount of textual data, prolonged training time, and considerable computational power. All these have made the training of new models a complicated procedure available only to selected research labs over the world, as oftentimes researchers lack the necessary resources to train their own models (our interest in historical language poses a particular challenge in this regard, as historical texts are usually much smaller in size).

To mitigate these requirements, speed up the training process, and increase the usability of these models, *fine-tuning* was developed. Using fine-tuning, models that were already trained (now called pre-trained models) are continued to be trained, albeit on much smaller data and sometimes on a different task (pre-trained models are usually trained on standard word and sentence prediction tasks).

This two-step training setup was found to greatly improve the state-of-the-art performance in many tasks and today, the use of fine-tuning in contextualized models has become the dominant paradigm in NLP (Howard and Ruder, 2018; Devlin et al., 2019; Merchant et al., 2020). Importantly, it was found that fine-tuning, henceforth FT, can *transfer* to other tasks and languages and thus improve performance on tasks and datasets it was not trained on (Peters et al., 2019; Khashabi et al., 2020), presumably because of shared information that is required to process these different tasks. In this paper, we aim to leverage the transfer capabilities and test whether FT a contextualized model on a range of NLP tasks improves its performance to detect LSC.

3 Method

Our aim is to test whether a state-of-the-art method for detecting lexical semantic change, based on pre-trained contextual embeddings, can be improved by adding fine-tuned layers. Therefore, we start with a BERT model and detect semantic change following Kutuzov and Giulianelli (2020). We chose their model as it was the best-performing system in English in the post-evaluation stage of SemEval-2020 Task 1. Next, we add information from fine-tuning for tasks that go beyond a masked language model objective. We include tasks aimed to capture both low-level linguistic information like part-of-speech tagging, as well as higher-level (semantic) information such as sentiment analysis, linguistic inference, and machine reading comprehension.

3.1 LSC method

The task of detecting lexical semantic change can be described as the following: given two corpora C_1 and C_2 from time periods T_1 and T_2, as well as a set of target words, detect which words have changed between T_1 and T_2 as evidenced in C_1 and C_2. This is a special case of the general LSC problem which includes arbitrarily many time periods $T_1, \ldots, T_N$.

Following Kutuzov and Giulianelli (2020), we use a pre-trained BERT base model to generate the contextualized embeddings of each occurrence of the target words in C_1 in C_2, resulting in two corresponding embedding matrices U_w^{t1} and U_w^{t2}. Given these embedding matrices, we calculate the change scores of each target word in one of two ways: inverted cosine similarity over word prototypes (PRT); and average pairwise cosine distance between token embeddings (APD).

$$PRT(U_w^{t1}, U_w^{t2}) = \frac{1}{d(\frac{\sum_{x_i \in U_w^{t1}} x_i}{N_w^{t1}}, \frac{\sum_{x_j \in U_w^{t2}} x_j}{N_w^{t2}})} \quad (1)$$

$$APD(U_w^{t1}, U_w^{t2}) = \frac{1}{N_w^{t1} N_w^{t2}} \sum_{x_i \in U_w^{t1}, x_j \in U_w^{t2}} d(x_i, x_j) \quad (2)$$

N_w^{t1}, N_w^{t2} stands for the number of occurrences of w in $T1$ and $T2$. d is the cosine similarity. For both methods, higher values suggest a larger semantic change.

3.2 Fine-tuning

The main contribution of this paper is the injection of richer meanings into contextualized embeddings using fine-tuning. Our fine-tuned models are derived mostly from adapters (Pfeiffer et al., 2020; Poth et al., 2021), which are trained layers that can be integrated directly into transformer-based models (the most popular type of contextualized models) in order to perform different tasks. Using adapters enabled us to speed up our experiments as they are readily available[2] and can be seamlessly integrated into the tested models.

In addition to using adapters, we also fine-tune two models locally on sentiment classification and part-of-speech tagging in order to compare the performance of fine-tuned models with adapter-based models. For sentiment classification, we use the sst2 dataset (Pang and Lee, 2004) while for part-of-speech tagging we use CoNLL2003 (Tjong Kim Sang and De Meulder, 2003). Since there is no test set with gold labels for sst2, we randomly sample 30% of the data from the validation set as a test set. The accuracy of the fine-tuned models on the test set is 0.908 for sentiment analysis and 0.931 for part-of-speech tagging. We use the BERT-base-uncased model for all our experiments, both with adaptors and local FT. Table 1 details the FT tasks we used.

Task/Model	Type
natural language inference (nli)	pf
machine reading comprehension (reading compre)	pf
sentiment (sst2)	pf
sentiment (sst2-pfeiffer)	pfeiffer
sentiment (sst2-hously)	hously
semantic textual similarity	pf
linguistic acceptability	pf
grammatical error correction (error detect)	pf
semantic tagging	pf
named entity recognition (ner)	pf
part-of-speech tagging (pos)	pf
phrase chunking	pf
sentiment (sst2-fine-tune)	fine-tuned
part-of-speech tagging (pos-fine-tune)	fine-tuned

Table 1: Fine-tuned models & tasks for adapters (upper) and locally trained FT (lower). Type refers to adaptors trained by Poth et al. (2021)(pf), Peiffer and Hously. Task abbreviations in parentheses.

[2] https://adapterhub.ml/

4 Evaluation

For our experiments, we use a standard evaluation dataset for LSC.

4.1 Evaluation data

We use the English dataset of SemEval-2020 Task 1 (Schlechtweg et al., 2020) for unsupervised lexical semantic change detection. The task was the first of its kind to provide manually annotated gold data for the purpose of fair and comparable evaluation of methods for LSC. The task consists of two sub-tasks aimed to measure change between two time-specific corpora C_1 and C_2:

Binary Classification: for a set of target words, decide which words lost or gained sense(s) between C_1 and C_2, and which did not.

Ranking: rank a set of target words according to their degree of LSC between C_1 and C_2.

These tasks are related but complementary. The ranking task measures the degree of change and takes into consideration lost or gained senses, but also includes changes in existing senses (e.g., by means of broadening or narrowing) which the binary classification task does not consider.

The English dataset of SemEval consists of 37 target words derived from the Clean Corpus of Historical American English (CCOHA) (Davies, 2012; Alatrash et al., 2020). The two 50-year periods are $C_1 = 1810{-}1860$ and $C_2 = 1960{-}2010$ from which each target word has a set of 100 randomly sampled sentences. These sentences have been compared by human annotators and ranked on a scale for lexical semantic change. Based on the outcome of the roughly 29,000 human judgments, words are classified as changing or stable and assigned a change degree. The process is described in detail by Schlechtweg et al. (2021)

4.2 Evaluation Metrics

We use three evaluation metrics.

Spearman correlation is used to compute the rank correlation between model predictions and gold labels in the ranking task.

AUC & ROC are used to evaluate the impact of different thresholds on the model performance in the binary classification task.

4.3 Validation through permutation

The work presented in this paper suggests that certain combinations of FT tasks improve LSC detection for both ranking and classification tasks. Importantly, these combinations are chosen based on improved performance in the SemEval tasks. Ideally, we would consider this as the training set and then test the chosen combinations on a held-out dataset to examine if similar gains are acquired. However, such a test set is lacking, and cannot be constructed via standard train-test splits from the 37 words, from which only 17 have changed.[3] We propose permutation tests to mitigate this shortcoming, and to enable us to draw reliable conclusions from our study despite this limitation.

For both tasks, we evaluate the probability that the best combinations we report were found by chance. We conduct a permutation analysis and generate artificial FT task scores that are based on the distributions of the existing FT values from the 14 FT models (Table 1). We then compute the relevant evaluation metric (Spearman or AUC) for each artificial FT, and repeat the process 100,000 times, creating a distribution of ranking or classification performance scores. We then compute the proportion of times that the artificial random FT performed better than our best combinations, in the form of a p-value for our chosen combinations. Ultimately, this allows us to test the statistical significance of our results, and evaluate how likely it is that our best combinations were found by chance.

5 Experiments

Two models are used as baselines, relative to which we test if adding FT (either adaptors or local FT) improves performance.

We choose the best combination of FT tasks by analyzing the results of the *ranking task* and then test it on the *binary classification task* using a modified decision criterion (i.e., ROC analysis). Because the two tasks are different, this allows us to use the latter as an ad-hoc evaluation test.

5.1 Baselines

We make use of two rather different baseline models that are used together with the fine-tuning. Kutuzov and Giulianelli (2020) whose model scored

[3] Under these conditions it was also not feasible to conduct a systematic regression analysis, which would have lacked the statistical rigor to reach reliable results.

the highest in the English part of SemEval (henceforth BERT), and HistBERT (Qiu and Xu, 2022) that provide the only contextualized model pretrained on historical English.[4] Together, they provide complementing baselines to test our research hypothesis.

We also compute p-values from the permutation tests for the two tasks (see Section 4.3), and for each method (APD and PRT).

5.2 Ranking task

Given a fine-tuning task FT_i, we obtain two embedding matrices U_w^{t1} and U_w^{t2} for each target word. We use these embedding matrices to calculate the semantic change score of a target word by means of APD and PRT. Once we have the change scores for all target words, we produce a ranking of the words. We then measure the performance as Spearman correlation of the change score ranks compared to the gold ranking. This is illustrated in the following formula, where P_{ind} stands for the performance of the individual task. $Score$ is the scoring function (AUC or Spearman correlation).

$$P_{ind} = Score(FT_i, gold) \quad (3)$$

There are in total $\text{FT}_1, \ldots, \text{FT}_{14}$ fine-tuning tasks. In addition to their individual performance, we are interested if they add complementary information, and therefore want to measure their combined performance. We thus enumerate all the possible combinations of the tasks. For each combination, we average the change scores produced by each participating FT_i. For instance, we can combine change scores derived from Natural Language Inference and Named Entity Recognition by averaging the scores for each target. There are in total FT_{14}^1 + $\text{FT4}_{14}^2 + \ldots +$ FT_{14}^{13} + FT_{14}^{14} combinations. We measured the performance of each combination by means of its Spearman correlation. This is illustrated in the following formula, where c is the combination of different tasks.

$$P_c = Score\left(\frac{1}{|c|}\sum_{i \in c}(FT_i), gold\right) \quad (4)$$

5.3 Binary Classification task

For the binary classification task, we calculate the AUC score of each FT_i. One advantage of AUC score over accuracy is that we do not need to define the threshold to determine a word changes or not, given the change scores derived from PRT and APD are continuous values

We carry out two experiments here: 1) testing the best models found in the ranking task on the binary classification task, and 2) examining the performance of individual FT_i as well as combined models. Our motivation for the first experiment is that we want to evaluate our best models from a new perspective given that the ranking and classification tasks feature different task profiles. In the second experiment, we focus on the combination effect, and take the more challenging case, examining the FTs with the highest as individual FT tasks.

6 Results

6.1 Ranking task results

We begin by reporting the results of individual FT_i. The results are presented in the first two columns of Table 2. For PRT, the range of correlation between (the ranking produced by) each FT_i and the gold rank is 0.427 – 0.076. The baseline results, as well as the best and worst combined results, are presented in Table 3. Interestingly, the maximum baseline result (BERT = 0.423) is lower than that of two individual FTs, namely sst2_hously and snli. However, the five best models (0.531 – 0.502) all rank above that of the highest FT by a wide margin, and significantly higher than the BERT baseline and HistBERT baseline (0.264).

For APD, individual FT_is range (0.694 – 0.086) with one FT having a negative correlation of 0.211. Here, the maximum baseline (BERT) is marginally higher than any individual FT with a correlation value of 0.706. The five best models all rank higher (0.723 – 0.721) and are statistically significant.

We conclude that combining different fine-tuned tasks increases model performance with more performance gains in PRT than in APD.

We also find that not every combination leads to an improvement. Some tasks/task combinations yield lower performance, exemplified by part-of-speech fine-tune and sentiment fine-tune in APD, histBERT-full, and the locally fine-tuned models in PRT. Their correlation scores are much lower than the scores provided by baseline BERT.

The random permutations where we average

[4] There are four versions: HistBERT-prototype, HistBERT-5, HistBERT-10, and HistBERT-full. They differ in the size and time period of the training data. In this study, we report the averaged scores of the four models.

Method	Rank-PRT	Rank-APD	AUC-PRT	AUC-APD
ner	0.218	0.285	**0.688***	0.604
nli	**0.427****	0.634	0.646	0.714
pos	0.205	0.205	0.634	0.536
error detect	0.352	0.593	0.634	0.670
linguistic acceptability	0.364	0.622	0.628	0.676
phrase chunking	0.076	0.185	0.622	0.622
pos-fine-tune	0.277	0.087	0.634	0.643
reading compre	0.416	0.636	0.670	0.696
semantic tagging	0.265	0.255	0.649	0.631
sst2	0.422	0.608	0.658	0.664
sst2-hously	**0.435****	0.627	**0.682***	0.685
sst2-fine-tune	0.123	0.210	**0.673***	0.417
sst2-pfeiffer	0.391	0.459	0.631	0.613
textual similarity	0.378	0.694	0.661	**0.741****
BERT	0.423	**0.706**	0.673	0.717
HistBERT-ave	0.264	0.441	0.657	0.659

Table 2: Spearman correlations and AUC scores for different FTs on the ranking and classification tasks. Best FTs in bold. Statistical significance marks *, **, ***: for p-values<.05, .01, .001, respectively.

Method	**Combination(s)**	**Correlation**
APD best 5	BERT, nli, textual similarity, error detect , pos-fine-tune	0.723***
	BERT, sst2, error detect, pos-fine-tune	0.722***
	sst2, textual similarity, error detect	0.722***
	BERT, sst2, textual similarity, error detect	0.721***
	BERT, textual similarity, error detect (,,0.00015)	0.721***
APD baseline	BERT	0.706
	BERT, random scores	0.462
	HistBERT (averaged)	0.441
PRT best 5	nli, pos-fine-tune, sst2-fine-tune	0.531**
	nli, sst2-pfeiffer, pos-fine-tune, ner, sst2-fine-tune	0.515*
	BERT, nli, sst2-pfeiffer, ner, sst2-fine-tune	0.503*
	nli, sst2-pfeiffer, pos-fine-tune, sst2-fine-tune	0.503*
	nli, reading compre, sst2-pfeiffer, histBERT-10, pos-fine-tune, sst2-fine-tune	0.502*
PRT baseline	BERT	0.423
	BERT, random scores	0.336
	HistBERT (averaged)	0.264

Table 3: Ranking results for 5 best FT combinations, APD and PRT. p-values as reported in Table 2.

BERT with scores sampled from the overall score distribution, corroborate our findings. For PRT and 100,000 runs, we get a mean correlation of 0.336 and standard deviation of 0.107, and less than one per 100 randomly sampled scores perform better than the best PRT (0.531). This makes our results highly significant statistically (p-value<0.01). For APD, the corresponding values are 0.462 and a standard deviation of 0.147. Less than 2 per 1000 randomly sampled scores perform better.

6.2 Classification task results

Also for the binary classification task, we begin by reporting the results of individual FT_i. We report the AUC scores to avoid threshold selection. The results are shown in Table 2. We observe more variance in APD than in PRT in terms of model performance. For PRT, we observe that named entity recognition provides the highest AUC score (0.688) while phrase chunking generates the lowest performance (0.622). As for APD, textual similarity was the only FT to surpass the baseline and achieve statistical significance with AUC score of 0.741.

To test the performance of combinations, we take the top 5 best-performing models for PRT and APD separately. We then enumerate all possible combinations from them and report the results in Table 4. We find that combining different FTs improves task performance in PRT but not in APD: in PRT, we see a five percentage of AUC increase

Method	Combination(s)	AUC
APD best 3	nli, textual similarity	0.741*
	textual similarity, nli, BERT	0.738*
	textual similarity, BERT	0.735*
APD ind best	textual similarity	0.741*
PRT best 3	ner sst2-hously, sst2-fine-tune *(p-value = 0.054)*	0.732
	ner, sst2-fine-tune, reading compre *(p-value = 0.090)*	0.720
	ner, sst2-hously, sst2-fine-tune, reading compre *(p-value = 0.161)*	0.714
PRT ind best	ner *(p-value = 0.161)*	0.688

Table 4: Results of combining different FTs on the classification task. We present the best individual task performance (APD/PRT ind best) for comparisons. * means significant improvement over the BERT baseline.

when we combine Name Entity Recognition, sst2-hously and sst2-fine-tune (from 0.688 to 0.732). In APD, we do not observe performance gains over the individual FTs.

As the ranking and classification asks are curated differently (see Section 4.1), we test the transferability between the two tasks. We ask, are the best combinations for ranking also useful (i.e., can be transferred) for the purpose of classification? We apply the three best combinations found in Section 5.2 to the classification task, plot their ROC curves and calculate their AUC scores[5]. The results are shown in Figure 1. We find that the best models found in the ranking task outperform BERT baseline in the classification task. In PRT, the best model achieves an AUC score of 0.774 compared with the base model (0.673). In APD, though less obvious, the best model still performs better than the baseline (0.744 v.s. 0.717). While the ranking and classification tasks are designed for different purposes, this experiment suggests that the analysis of ranking results can guide the choice, and thus transfer, of models for the classification task.

7 Discussion and Conclusion

In this paper, we investigate our hypothesis that adding linguistic information to pre-trained language models by means of fine-tuning can lead to improved performance on unsupervised Lexical Semantic Change detection. We chose two classic LSC tasks, ranking and binary classification, from SemEval-2020 Task 1 (Schlechtweg et al., 2020). To simply and speed up the process of fine-tuning we used adaptors (Pfeiffer et al., 2020), that are pre-trained fine-tuning modules that can augment existing contextualized models and are readily and freely available in English.

First, we tried single FT tasks, which showed little or no improvement over the BERT baseline. Then we combined several FT tasks together by means of simple averaging and found considerable improvements.

Adding linguistic information, like part-of-speech, to a standard (masked) language model can offer additional information that improves the ability of the models to detect lexical semantic change. However, some kinds of information are adjacent to semantic change and therefore make the models capture change, but not change that falls into the class of semantic change. In future work, we will conduct in-depth analysis of the worst models, to see what information they capture instead and why this information seems to hurt the performance of the LSC model.

One deficiency of our results is that they are based on a small evaluation dataset, which means that the improvements we report could be attributed to chance (or model over-fitting). To mitigate this concern, and add scientific rigour to our analyses, all results were tested in permutation tests and are reported with their statistical significance values. Evaluation is also done with comparison to two strong baseline models, each of which provides different perspective to test our research hypothesis. Outperforming the Kutuzov and Giulianelli (2020) model suggests that enriching models with additional linguistic information is highly beneficial, and outperforming HistBERT supports the idea that this information cannot be gained even when a model is exclusively (pre-) trained on historical text. Overall our results clearly suggest that LSC models can be improved

[5]We also calculate the accuracy. However, the results are less clear than in the AUC scores. This might be attributed to the limited thresholds we create: using six thresholds cannot fully capture model performance compared to using AUC in this case.

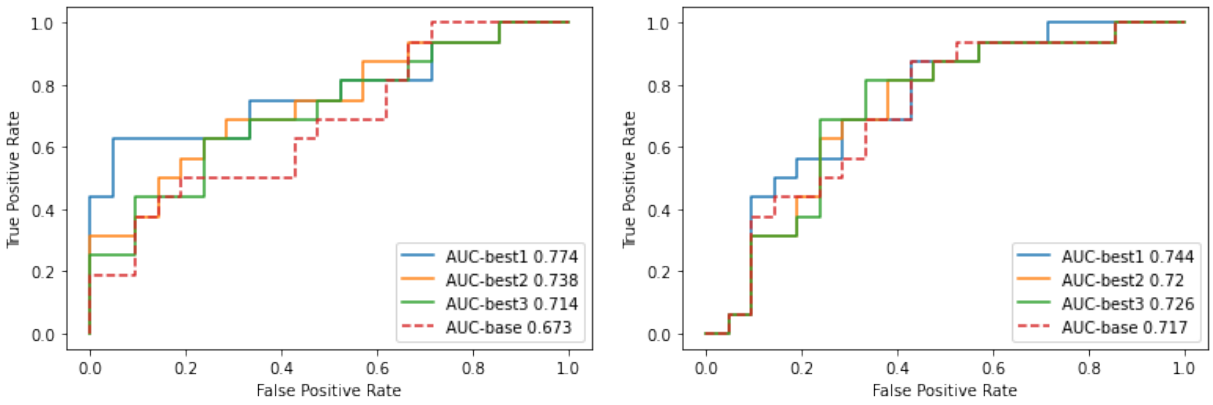

Figure 1: ROC curves of the best 3 models in the ranking task as they perform on the classification task. Left: PRT, right: APD. See Table 3 for combinations details.

dramatically with relatively simple steps of fine-tuning on a range of standard linguistic tasks.

We note that there are differences between the best performing FTs for the APD and PRT methods. Although certain FTs are shared (e.g., NLI and POS), others appear more systematically in APD or PRT. We interpret these differences as stemming from the differences in nature of the binary and classification problem. Each method enriches the baseline with different types of information and hence allows the model to capture slightly different aspects of LSC.

We propose that our conclusions are inline with how linguistic theory describes LSC which distinguishes different types of LSC, and emphasizes that changes are never 'general' but pertinent to certain aspects of meaning. Therefore, computationally analyzing words' meaning change using a "single ruler" as is done by current state-of-the-art LSC models, may be insufficient to describe the richness and diversity of change. We believe our findings provides an inroad for extending the capacity of LSC models, and encouraging future research in this direction,

This is but the first step in exploring the potential of using FT to enrich and improve contextualized models of LSC. In our further work we will corroborate these findings more rigorously: extending these to other languages, testing the generalizability of the chosen FT tasks across LSC models, and test our approach in the discovery of new cases of words that change their meaning to go beyond small set of examples.

References

Reem Alatrash, Dominik Schlechtweg, Jonas Kuhn, and Sabine Schulte im Walde. 2020. CCOHA: Clean Corpus of Historical American English. In *Proceedings of the Twelfth International Conference on Language Resources and Evaluation (LREC'20)*. European Language Resources Association (ELRA).

Rabab Alkhalifa, Elena Kochkina, and Arkaitz Zubiaga. 2023. Building for tomorrow: Assessing the temporal persistence of text classifiers. *Information Processing & Management*, 60(2):103200.

Nikolay Arefyev and Vasily Zhikov. 2020. BOS at SemEval-2020 Task 1: Word Sense Induction via Lexical Substitution for Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Francesco Barbieri, Jose Camacho-Collados, Bhuwan Dhingra, Luis Espinosa-Anke, Elena Gribovskaya, Angeliki Lazaridou, Daniel Loureiro, and Leonardo Neves, editors. 2022. https://aclanthology.org/2022.evonlp-1.0 *Proceedings of the The First Workshop on Ever Evolving NLP (EvoNLP)*. Association for Computational Linguistics, Abu Dhabi, United Arab Emirates (Hybrid).

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.

Mark Davies. 2012. Expanding Horizons in Historical Linguistics with the 400-Million Word Corpus of Historical American English. *Corpora*, 7(2):121–157.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. https://doi.org/10.18653/v1/N19-1423 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.

Haim Dubossarsky, Daphna Weinshall, and Eitan Grossman. 2016. Verbs change more than nouns: A bottom up computational approach to semantic change. *Lingue e Linguaggio*, 1:5–25.

Amaru Cuba Gyllensten, Evangelia Gogoulou, Ariel Ekgren, and Magnus Sahlgren. 2020. SenseCluster at SemEval-2020 Task 1: Unsupervised Lexical Semantic Change Detection. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Jeremy Howard and Sebastian Ruder. 2018. https://doi.org/10.18653/v1/P18-1031 Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 328–339, Melbourne, Australia. Association for Computational Linguistics.

Daphna Keidar, Andreas Opedal, Zhijing Jin, and Mrinmaya Sachan. 2022. Slangvolution: A causal analysis of semantic change and frequency dynamics in slang. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*, volume 1, pages 1422–1442. Association for Computational Linguistics.

Daniel Khashabi, Sewon Min, Tushar Khot, Ashish Sabharwal, Oyvind Tafjord, Peter Clark, and Hannaneh Hajishirzi. 2020. https://doi.org/10.18653/v1/2020.findings-emnlp.171 UNIFIEDQA: Crossing Format Boundaries with a Single QA System. In *ACL*, pages 1896–1907.

Andrey Kutuzov and Mario Giulianelli. 2020. https://doi.org/10.18653/v1/2020.semeval-1.14 UiO-UvA at SemEval-2020 task 1: Contextualised embeddings for lexical semantic change detection. In *Proceedings of the Fourteenth Workshop on Semantic Evaluation*, pages 126–134, Barcelona (online). International Committee for Computational Linguistics.

Andrey Kutuzov, Lilja Øvrelid, Terrence Szymanski, and Erik Velldal. 2018. Diachronic word embeddings and semantic shifts: A survey. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1384–1397, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Andrey Kutuzov, Lidia Pivovarova, et al. 2021. Rushifteval: a shared task on semantic shift detection for russian. In *Computational linguistics and intellectual technologies: Papers from the annual conference Dialogue*. Redkollegija sbornika.

Andrey Kutuzov, Samia Touileb, Petter Mæhlum, Tita Enstad, and Alexandra Wittemann. 2022. https://aclanthology.org/2022.lrec-1.274 NorDiaChange: Diachronic semantic change dataset for Norwegian. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 2563–2572, Marseille, France. European Language Resources Association.

Amil Merchant, Elahe Rahimtoroghi, Ellie Pavlick, and Ian Tenney. 2020. https://doi.org/10.18653/v1/2020.blackboxnlp-1.4 What happens to BERT embeddings during fine-tuning? In *Proceedings of the Third BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pages 33–44, Online. Association for Computational Linguistics.

Bo Pang and Lillian Lee. 2004. https://doi.org/10.3115/1218955.1218990 A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 271–278, Barcelona, Spain.

Matthew E. Peters, Sebastian Ruder, and Noah A. Smith. 2019. https://doi.org/10.18653/v1/W19-4302 To tune or not to tune? adapting pretrained representations to diverse tasks. In *Proceedings of the 4th Workshop on Representation Learning for NLP (RepL4NLP-2019)*, pages 7–14, Florence, Italy. Association for Computational Linguistics.

Jonas Pfeiffer, Andreas Rücklé, Clifton Poth, Aishwarya Kamath, Ivan Vulić, Sebastian Ruder, Kyunghyun Cho, and Iryna Gurevych. 2020. https://www.aclweb.org/anthology/2020.emnlp-demos.7 Adapterhub: A framework for adapting transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP 2020): Systems Demonstrations*, pages 46–54, Online. Association for Computational Linguistics.

Clifton Poth, Jonas Pfeiffer, Andreas R"uckl'e, and Iryna Gurevych. 2021. https://aclanthology.org/2021.emnlp-main.827 What to pre-train on? Efficient intermediate task selection. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 10585–10605, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wen Qiu and Yang Xu. 2022. Histbert: A pretrained language model for diachronic lexical semantic analysis. *ArXiv*, abs/2202.03612.

Guy Rosin and Kira Radinsky. 2022. Temporal attention for language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1498–1508.

Dominik Schlechtweg, Stefanie Eckmann, Enrico Santus, Sabine Schulte im Walde, and Daniel Hole. 2017. German in flux: Detecting metaphoric change via word entropy. In *Proceedings of the 21st Conference on Computational Natural Language Learning*, pages 354–367, Vancouver, Canada.

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 international Conference on Computational Linguistics (COLING)*.

Dominik Schlechtweg, Nina Tahmasebi, Simon Hengchen, Haim Dubossarsky, and Barbara McGillivray. 2021. https://doi.org/10.18653/v1/2021.emnlp-main.567 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. Association for Computational Linguistics.

Nina Tahmasebi, Lars Borin, and Adam Jatowt. 2021. https://doi.org/10.5281/zenodo.5040302 Survey of computational approaches to lexical semantic change detection.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. https://aclanthology.org/W03-0419 Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Frank D. Zamora-Reina and Felipe Bravo-Marquez. 2020. DCC-Uchile at SemEval-2020 Task 1: Temporal Referencing Word Embeddings. In *Proceedings of the 14th International Workshop on Semantic Evaluation*, Barcelona, Spain. Association for Computational Linguistics.

Frank D. Zamora-Reina, Felipe Bravo-Marquez, and Dominik Schlechtweg. 2022. https://aclanthology.org/2022.lchange-1.16/ 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.